Multi-level Models

In this module, you'll be familiarized with **multi-level models** as an approach for modeling *nested data*. You'll frequently encounter nested data structures, for example:

- Predicting **student** college admissions, where students are drawn from different *high schools*
- Modeling **patient** health outcomes, where patients are drawn from different *hospitals*
- Estimating **faculty** salaries, where the faculty are drawm from different *departments*

In each example above the **observations** (**students**, **patients**, **faculty**) belong to different *groups* (*schools*, *hospitals*, *departments*). These data can be described as **nested** because each observation comes from within a group (and we believe these groups to be related to the outcome). As you can imagine, there could be some effect at the *group level*, as well as the individual level. For example, a student's college acceptance may depend on *individual predictors*, such as their GPA, number of volunteer activities, and other factors. However, their admissions status may also depend on which *school* they belong to, based on the reputation of that school, financial aid, or other factors. In this module, we'll explore various (introductory) ways to handle nested data.

Note, while this example uses a linear model (with the `lmer`) function, these approaches can be easily extended to *generalized linear models* using the `glmer` function.

Vocabulary

There are a variety of different terms that statisticians use to refer to modeling nested data. Confusingly, *many terms* may refer to the *same procedure*, and many people may use the *same term* to refer to *different procedures*. The vocabulary introduced here largely comes from this cannonical text.

- **Multi-level models**: this term refers to modeling strategies for working with nested data. Using one of many possible approaches, these appropriately handle the fact that variables may exist at multiple levels (i.e., a dataset may describe *individuals* as well as the *cities* they come from). These are also often refered to as **hierarchical models**, because the data exist in a hierarchical structure (i.e., people within cities)
- **Fixed effects**: One component of a multi-level model is the set of *fixed effects* that are estimated. These, in short, are the betas (coefficients) you are familiar with estimating (i.e., each β value in this formula):

  $$\hat{y} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$$

  These effects are considered *fixed* because they are *constant* across individuals (observations) in the dataset, and are commonly estimated through least-squares (or, more genearlly, maximum likelihood methods). they While this text prefers to refer to these as **constant slopes** (and intentionally avoids the term *fixed effects*), you will encounter it commonly in other literature.
- **Random effects**: This an an umbrella term referred to ways in which you can incorporate information about *group level variation* in your model (i.e., variation across the *cities* that *individuals* live within). Broadly speaking, one may expect the variation across groups to vary *randomly*, and multi-level models allow you to incorporate that information in various ways. In the section below, we'll describe the ways in which this variation can be built into your model

Group level variation

As described above, your outcome variable (y) may vary based on the *group* to which each observation belongs. Linear models are comprised of two components: the *intercept* and *slope*. Appropriately, you may expect an individual's group to determine their baseline *intercept*, or an associated *slope*.

Varying Intercept

One type of **random effect** is to allow each group to determine the *intercept* for each observation. Using faculty salary data as an example, it may be the case that each *department* has a different baseline salary for each faculty member, but the averge increase in salary for each year of experience is consistent across the *University*. This could be written as a *mixed effects* model as follows:

$$y_i = \alpha_{j[i]} + \beta x_i$$

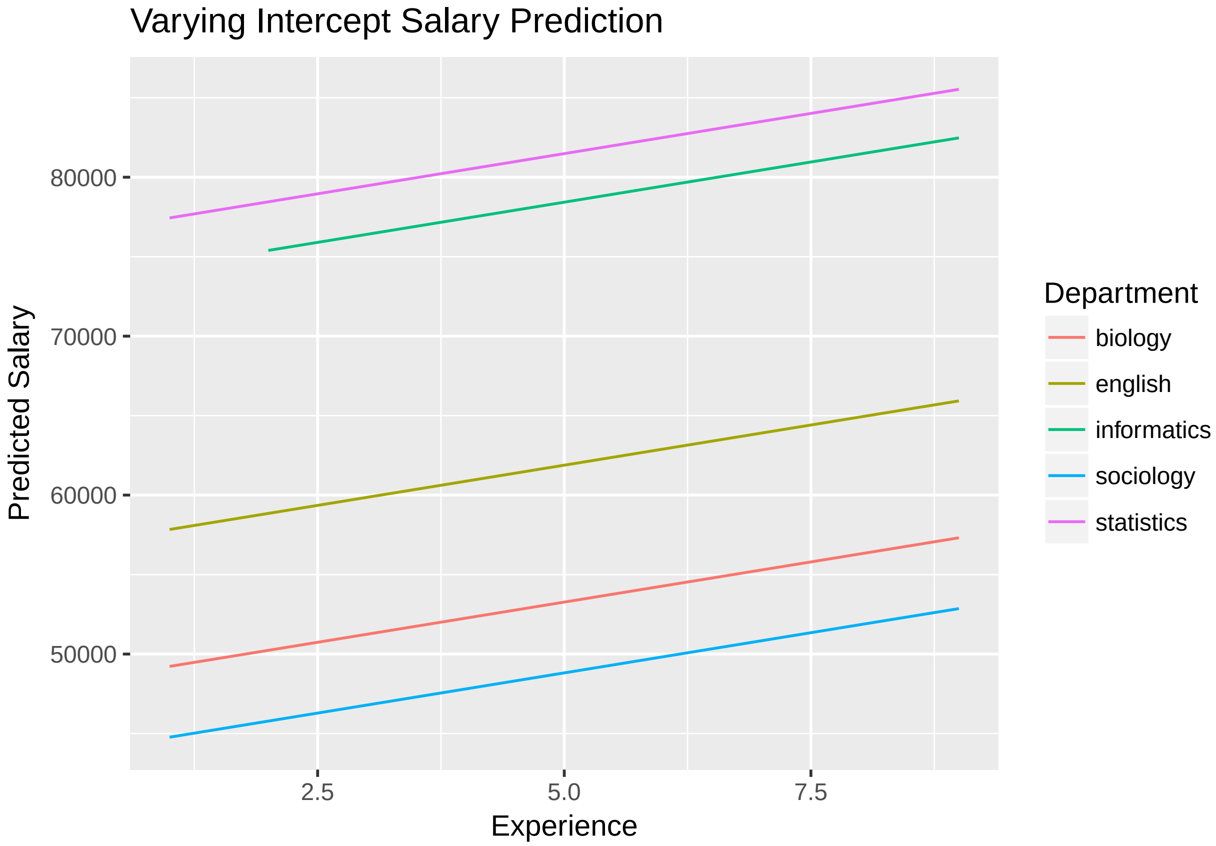
In the above formula, the vector of **fixed effects** (constant slopes) is represented by the term β. The **random intercept**, for individual i group j is denoted as $\alpha_{j[i]}$. Applying this to the simulated faculty dataset (from `faculty-data.R`), the formula be written as:

$$salary_i = department_{j[i]} + 1500 * experience_i$$

In this example, a faculty member's salary depends on the **base departmnet salary** of department j that person i belongs to ($department_{j[i]}$) plus \$1500 times the amount of experience of individual i ($1500 * experience_i$). This model is easily implement in R using the `lme4` package:

```r
# Model with varying intercept
m1 <- lmer(salary ~ experience + (1|department), data = df)
```

In the above code, the `lmer` function has a **fixed effect** on level of experience, and a **varying intercept** (i.e., *random effect* for intercept) for the department. The **predicted values** are shown here:



As you can see in the above graph, the predicted salary values increase linearly with experience because of the **fixed effect**. However, each department has a different intercept, because of the **random effect** associated with each department. This explains group level variation in slopes, and helps improve the accuracy of the model (based on our assumption of the faculty salary policy).

Varying slope

Another type of **random effect** is to allow each group to determine the *slope* for each observation. Using faculty salary data as an example, it may be the case that the University has a constant baseline salary, and each *department* has a different average incrase in salary for each year of experience. This could be written as a *mixed effects* model as follows:

$$y_i = \alpha + \beta_{j[i]} x_i$$

In the above formula, the vector of **random effects** (varying slopes) is represented by the term $\beta_{j[i]} x_i$. This retrieves the *slope* for group j, of which individual i is a member. The **constant intercept** across individuals is denoted as α. Applying this to some (hypothetical) faculty dataset, it could be written as:

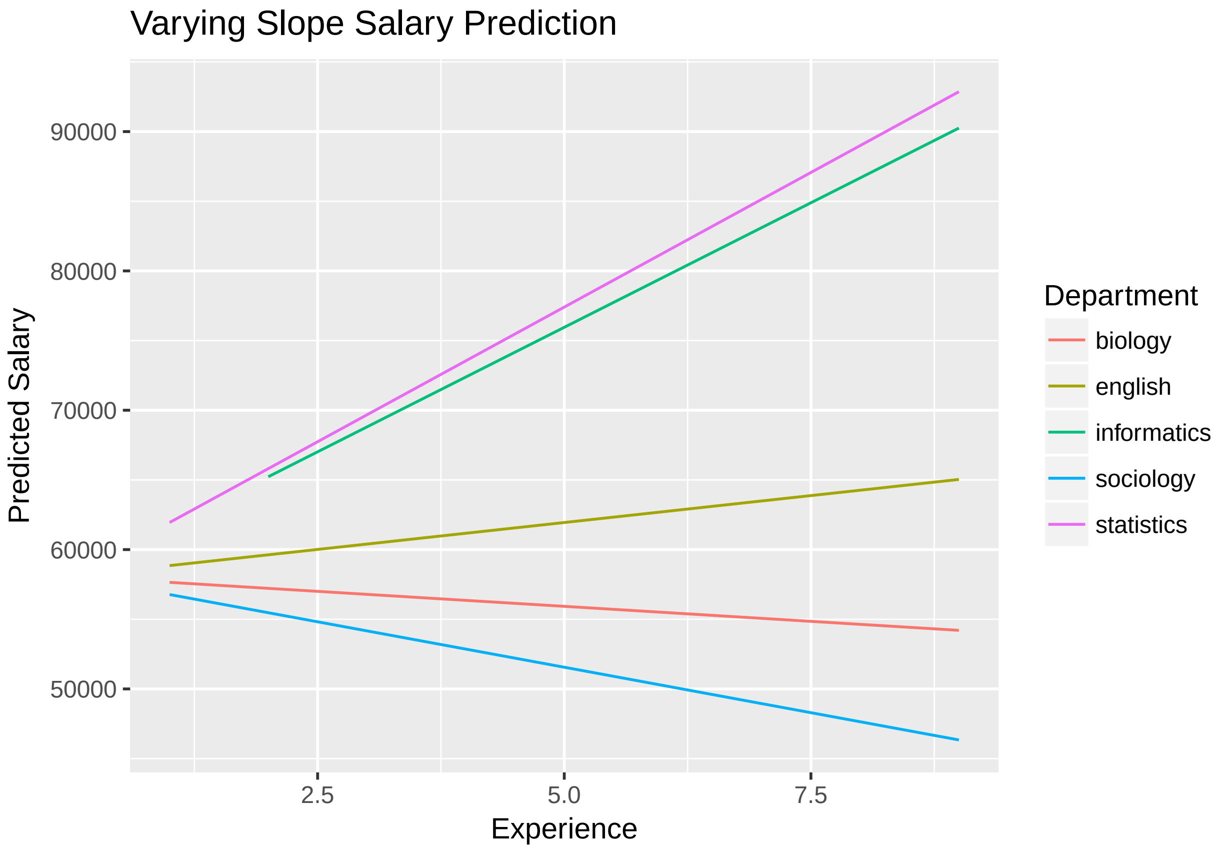
$$salary_i = 20000 + Raise_{j[i]} * experience_i$$

In this example, a faculty member's salary starts at $20,000 (regardless of department). Estimating their salary requires that you retrieve the estimated slope for departmet j which individual i belongs to ($Raise_{j[i]}$).

A faculty member's salary thus depends on the **base University salary**, and the slope (annual raise) associated with each deparment of department ($Raise_{j[i]}$) This model is easily implement in R using the `lme4` package:

```r
# Model with varying slope
m2 <- lmer(salary ~ experience + (0 + experience|department), data=df)
```

The syntax here is a bit more dense. As suggested in the documentation, `experience` is written to indicate a **fixed and random effect**. This can be thought of as "*random deviations from a fixed mean*" (source). Note, the `0` inside the random effect specification indiates that we **do not want a random intercept**. However, this model is quite strange – because it varies only slope (and not intercept) our predicted values do not follow the data well:



Varying slope and intercept

As you can imagine, we can also specify a model in which the slope and intercept both vary. Continuing with our example, this would imply each department has a different starting salary (*varying intercept*), **and** each department has a different raise associated with each year of experience (*varying slope*). That model can be described as follows:

$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_i$$

Or, using our salary data:

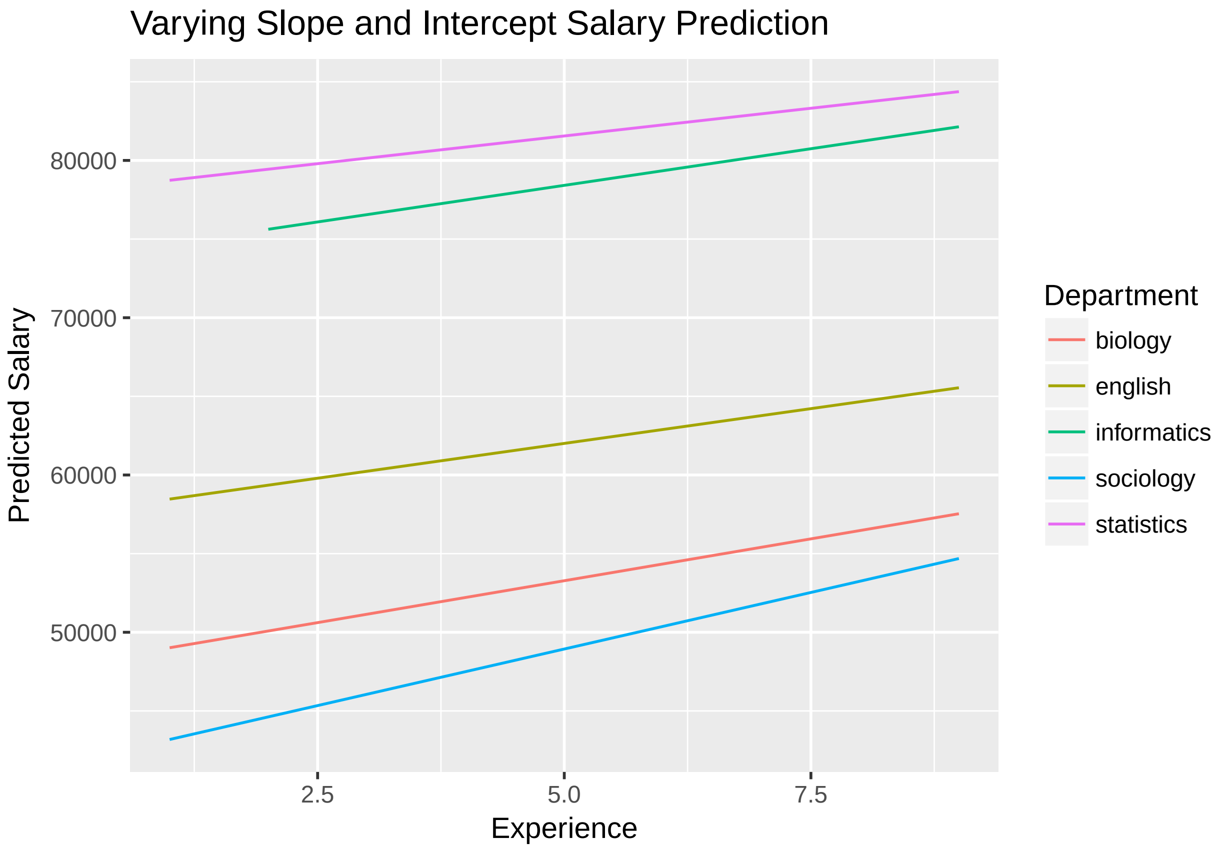
$$salary_i = base_{j[i]} + raise_{j[i]} x_i$$

Someone's salary thus depends on the *base salary of their department* ($base_{j[i]}$) as well as the annual raise associated with each year of experience *in their department* ($raise_{j[i]} x_i$).

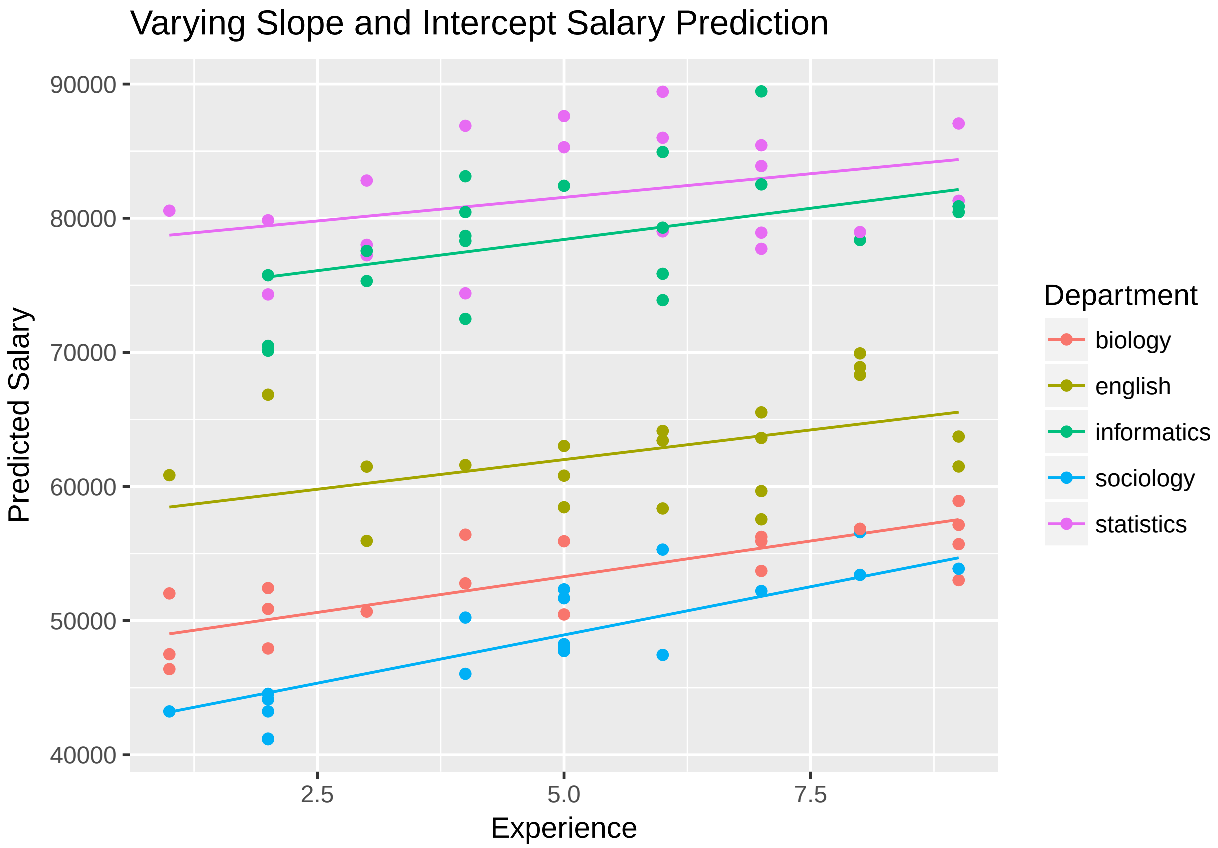
This is also easily implelented in R:

```r
# Model with varying slope and intercept
m3 <- lmer(salary ~ experience + (1 + experience|department), data=df)
```

The only difference between this model (`m3`) and the previous model (`m2`) is that the number `1` inside of the random effect specificaiton indicates that we want a random slope **and** a random intercept. As you might expect, the random slopes and intercepts are both captured in our estimates:



This (conceptually) fits our understanding of the university well. Plotting this alongside the data, we can see that this fits our data fairly well:



Interpretation

We'll use similar metrics for interpreting the direction, magnitude, and strength of relationship between our predictors and our outcome of interest. To retrieve coefficient values from an `lmer` model, use the `coef` function as shown here:

```
coef(m1)$department
```

```
##             (Intercept) experience
## biology        48212.63   1011.317
## english        56826.04   1011.317
## informatics    73372.25   1011.317
## sociology      43759.00   1011.317
## statistics     76428.32   1011.317
```

As you can see, the **intercept** varies by department for our first model, but the slope is constant. In our third model, both the slope and intercept varied:

```
coef(m3)$department
```

```
##             (Intercept) experience
## biology        47952.89  1064.8669
## english        57583.48   884.7232
## informatics    73760.02   930.9677
## sociology      41745.33  1438.1168
## statistics     78030.64   704.6838
```

We an interprete each of these numbers intuitively: someone the biology department is expected to have a base salary of 47952, and each year, their salary should increase by 1064.

There are a variety of approaches to estimating pvalues for mixed-models, and the "best practice" is debated (see `?pvalues` for a discussion). However, the authos of this book wrote the `arm` package to accompany their text, which includes helper functions for calculating standard errors of the estimates. They can be extracted using the following functions:

```
# Get standard errors for fixed effects
se.fixef(m1)

# Get standard errors for random effects
se.ranef(m1)
```

We can then compute confidence intervals as falling within **2 standard errors** of our point estimates. Correspondingly, if these estimates include 0, the relationship is not significant:

```
# Get coefficients for first model
coefs <- coef(m1)$department[,1]

# Get standard errors for random effects
ses <- se.ranef(m1)$department
upper <- coefs + 2*ses
lower <- coefs - 2*ses
bounds <- data.frame(lower, upper)
colnames(bounds) <- c('lower', 'upper')
bounds
```

```
##                lower    upper
## biology     46629.75 49795.51
## english     55243.17 58408.92
## informatics 71789.37 74955.13
## sociology   42176.12 45341.88
## statistics  74845.45 78011.20
```

We can then discuss the uncertainty of our (varying) intercepts. For example, we are 95% confident that the *true intercept* for the biology department falls between 46629 and 49795.