432 Class 23 Slides

thomaselove.github.io/432

2022-04-07

Today's R Packages

```r
library(janitor); library(here)
library(knitr); library(magrittr)
library(lme4)
library(arm)
library(broom); library(broom.mixed)
library(tidyverse)

theme_set(theme_bw())
```

An Introduction to Working with Hierarchical Data

- In a moment, we'll visit http://mfviz.com/hierarchical-models/.

There, we try to learn about nested (hierarchical) data on faculty salaries. For each subject (faculty member) in the data, we have information on their salary, department and years of experience.

- outcome: faculty salary (in $)
- predictor: years of experience
- group: department (five levels: Informatics, English, Sociology, Biology, Statistics)

We expect that salary (and the relationship between salary and years of experience) may be different depending on department, and every subject is in exactly one department.

Visual Explanation

We'll visit http://mfviz.com/hierarchical-models/ now to learn a bit about:

- Nested Data
- Linear Model on the Fixed Effects
- Adding Random Intercepts to the Fixed Effects Model
- Incorporating Random Slopes with a Constant Intercept
- Random Slope and Random Intercept

Fitting Hierarchical Models in R

We'll focus today on approaches using the `lme4` package, which can be used both for linear mixed models and for generalized linear mixed models.

- There are many, many ways to do this.
- The Generalized Linear Mixed Models FAQ at https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html describes lots of other options for fitting hierarchical models in R.

How The Data Were Simulated (From Github)

```r
# Parameters for generating faculty salary data
departments <- c('sociology', 'biology', 'english',
                 'informatics', 'statistics')
base.salaries <- c(40000, 50000, 60000, 70000, 80000)
annual.raises <- c(2000, 500, 500, 1700, 500)
faculty.per.dept <- 25
total.faculty <- faculty.per.dept * length(departments)
```

```r
# Generate tibble of faculty and (random) years of experience
set.seed(432)
ids <- 1:total.faculty
department <- rep(departments, faculty.per.dept)
experience <- floor(runif(total.faculty, 0, 10))
bases <- rep(base.salaries, faculty.per.dept) *
    runif(total.faculty, .9, 1.1) # noise
raises <- rep(annual.raises, faculty.per.dept) *
    runif(total.faculty, .9, 1.1) # noise
facsal <- tibble(ids, department, bases, experience, raises)
# Generate salaries (base + experience * raise)
facsal <- facsal %>%
    mutate(salary = bases + experience * raises,
           department = factor(department))
```

The `facsal` data

```
facsal
```

```
# A tibble: 125 x 6
     ids department   bases experience raises salary
   <int> <fct>        <dbl>      <dbl>  <dbl>  <dbl>
 1     1 sociology   39438.          2  1861. 43160.
 2     2 biology     46046.          0   493. 46046.
 3     3 english     63656.          9   458. 67776.
 4     4 informatics 67330.          1  1573. 68903.
 5     5 statistics  84116.          7   545. 87930.
 6     6 sociology   41626.          9  1871. 58463.
 7     7 biology     54687.          7   521. 58335.
 8     8 english     64477.          2   468. 65413.
 9     9 informatics 64456.          6  1722. 74787.
10    10 statistics  87841.          9   548. 92774.
# ... with 115 more rows
```

Linear Model (no grouping by department)

```
m0 <- lm(salary ~ experience, data = facsal)

tidy(m0, conf.int = TRUE) %>%
    select(term, estimate, std.error,
           conf.low, conf.high) %>%
    kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	56100.96	2285.57	51576.81	60625.10
experience	1772.01	412.96	954.58	2589.44

Linear Model Summary

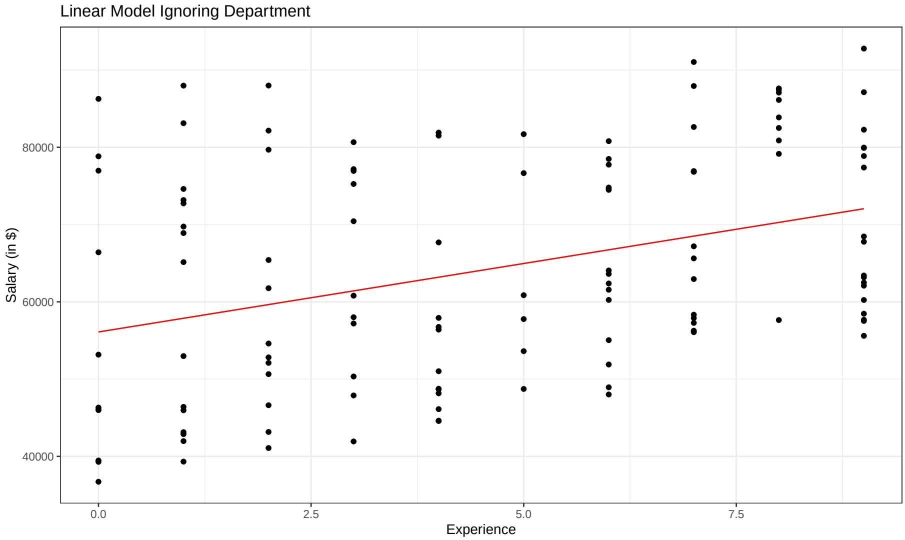
```r
glance(m0) %>%
    select(r.squared, adj.r.squared, sigma, AIC, BIC) %>%
    kable(digits = c(3, 3, 2, 2, 2))
```

r.squared	adj.r.squared	sigma	AIC	BIC
0.13	0.123	13757.72	2741.06	2749.54

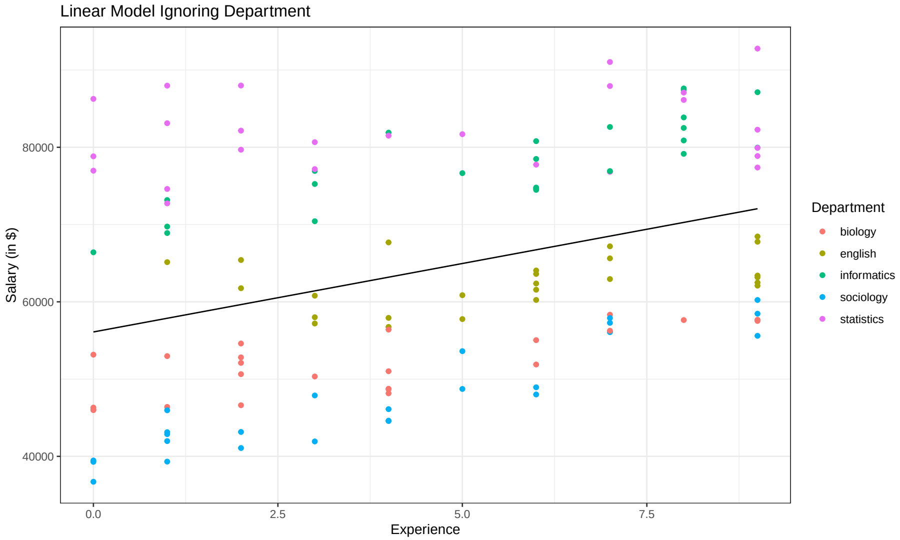
```r
facsal$simple_model_preds <- predict(m0)

head(predict(m0))
```

```
       1        2        3        4        5        6 
59644.98 56100.96 72049.05 57872.97 68505.03 72049.05 
```

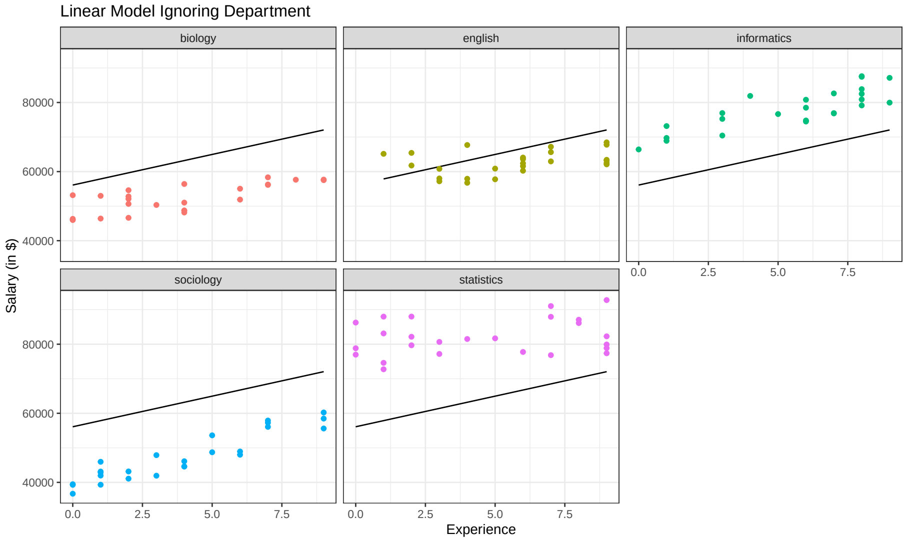
Plotting the m0 predictions and the data



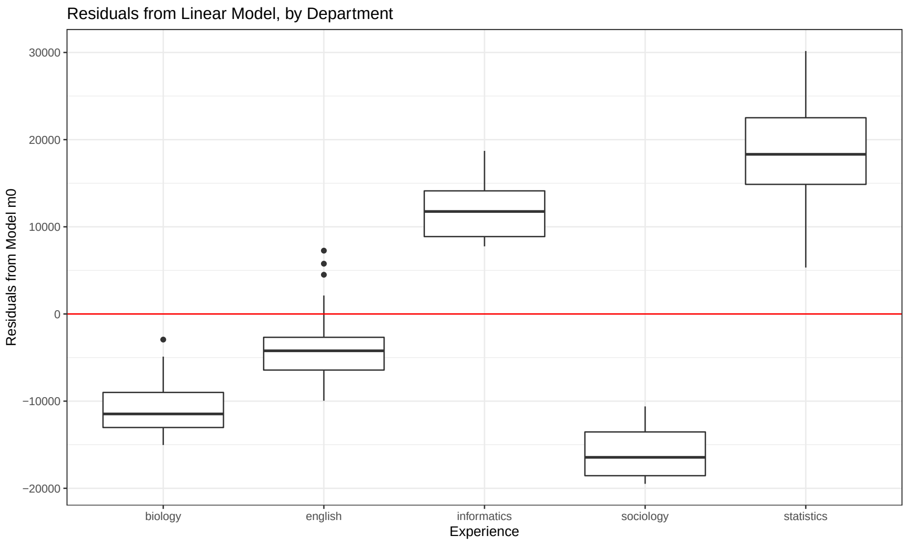
m0 predictions with Department indicators



m0 predictions and faceted results by Department



Plot of `m0` Residuals by Department



Let the intercepts vary

Model incorporating varying intercepts by department

```
m1 <- lmer(salary ~ experience + (1 | department),
           data = facsal)
```

Varying Intercept Model

```
m1
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: salary ~ experience + (1 | department)
   Data: facsal
REML criterion at convergence: 2428.544
Random effects:
 Groups     Name        Std.Dev.
 department (Intercept) 14728
 Residual                4069
Number of obs: 125, groups:  department, 5
Fixed Effects:
(Intercept)   experience
      59056         1138
```

Tidied Coefficients (use warning = FALSE)

```
tidy(m1, conf.int = TRUE) %>%
    select(-std.error, -statistic) %>%
    kable(digits = 0)
```

effect	group	term	estimate	conf.low	conf.high
fixed	NA	(Intercept)	59056	46075	72037
fixed	NA	experience	1138	890	1387
ran_pars	department	sd__(Intercept)	14728	NA	NA
ran_pars	Residual	sd__Observation	4069	NA	NA

Summarizing model m1

```
glance(m1) %>%
    select(sigma, AIC, BIC, logLik, df.residual) %>%
    kable(digits = 2)
```

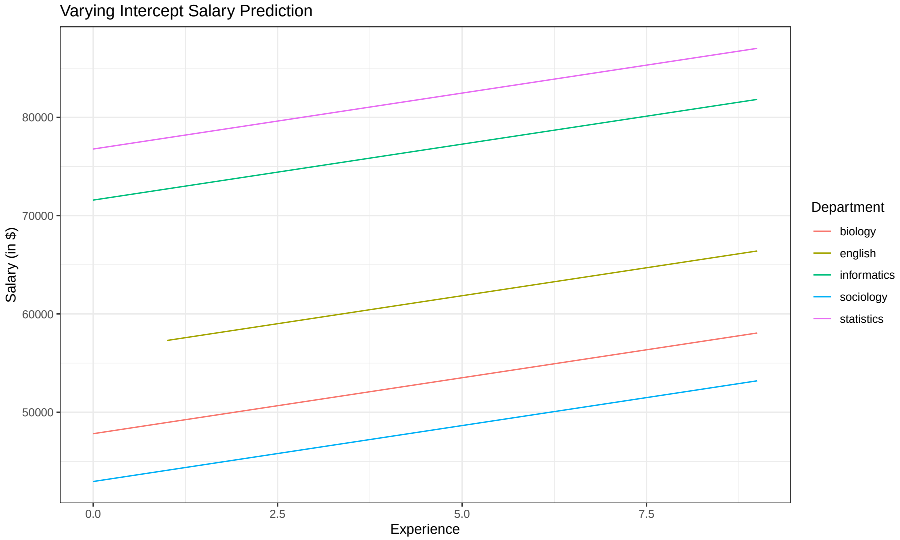
sigma	AIC	BIC	logLik	df.residual
4068.93	2436.54	2447.86	-1214.27	121

Saving the Model `m1` predictions

```r
facsal$random_intercept_preds <- predict(m1)

head(predict(m1))
```

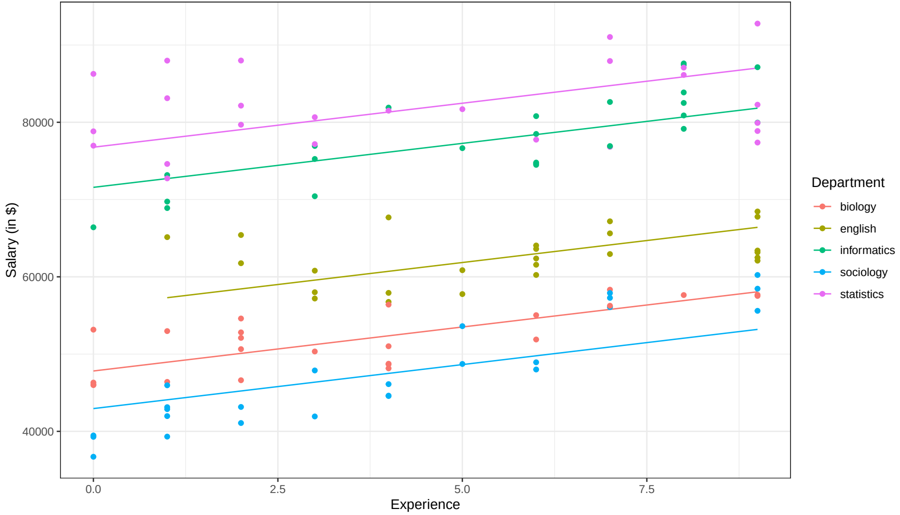
```
       1        2        3        4        5        6 
45226.95 47815.47 66405.51 72718.80 84743.65 53195.42 
```

Plotting the `m1` predictions without the data

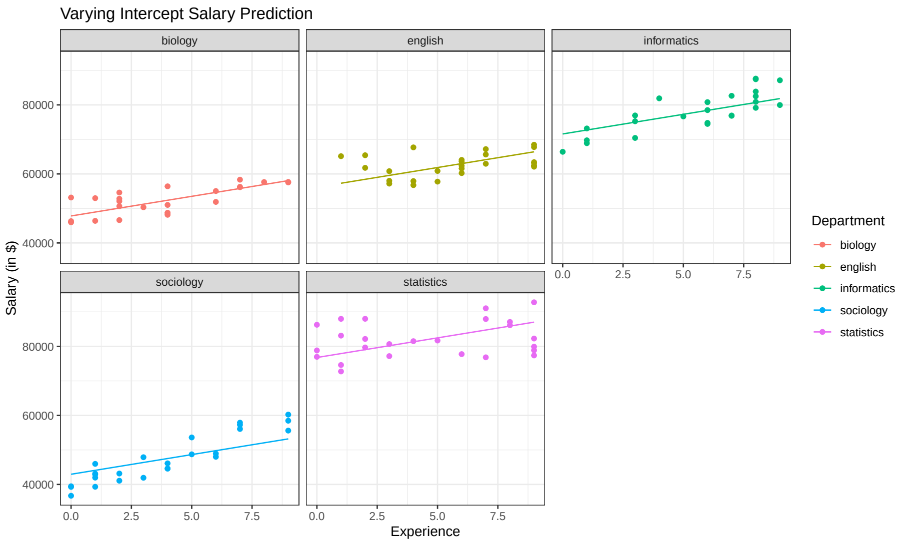


Plotting the `m1` predictions and the data

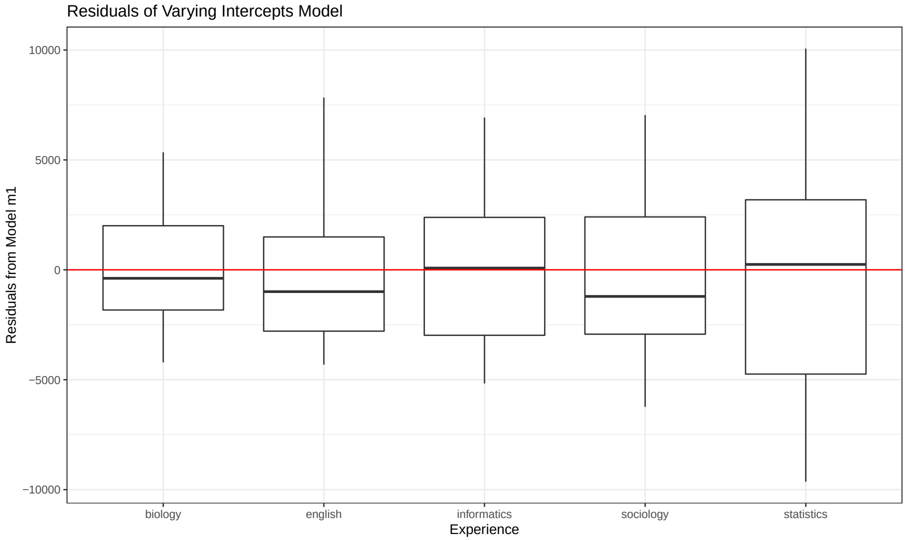
Varying Intercept Salary Prediction



`m1` predictions and the data, faceted by Department



Plot of `m1` Residuals by Department



Let the slopes vary

Model incorporating varying slopes by department

```
m2 <- lmer(salary ~ experience +
               (0 + experience | department),
           data = facsal)
```

Varying Slopes Model

```
m2
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
salary ~ experience + (0 + experience | department)
   Data: facsal
REML criterion at convergence: 2626.859
Random effects:
 Groups     Name       Std.Dev.
 department experience 2082
 Residual              9438
Number of obs: 125, groups:  department, 5
Fixed Effects:
(Intercept)   experience
      57705         1249
```

Tidied `m2` Coefficients (use warning = FALSE)

```
tidy(m2, conf.int = TRUE) %>%
    select(-std.error, -statistic) %>%
    kable(digits = 0)
```

effect	group	term	estimate	conf.low	conf.high
fixed	NA	(Intercept)	57705	54617	60793
fixed	NA	experience	1249	-661	3160
ran_pars	department	sd__experience	2082	NA	NA
ran_pars	Residual	sd__Observation	9438	NA	NA

Summarizing model m2

```
glance(m2) %>%
    select(sigma, AIC, BIC, logLik, df.residual) %>%
    kable(digits = 2)
```

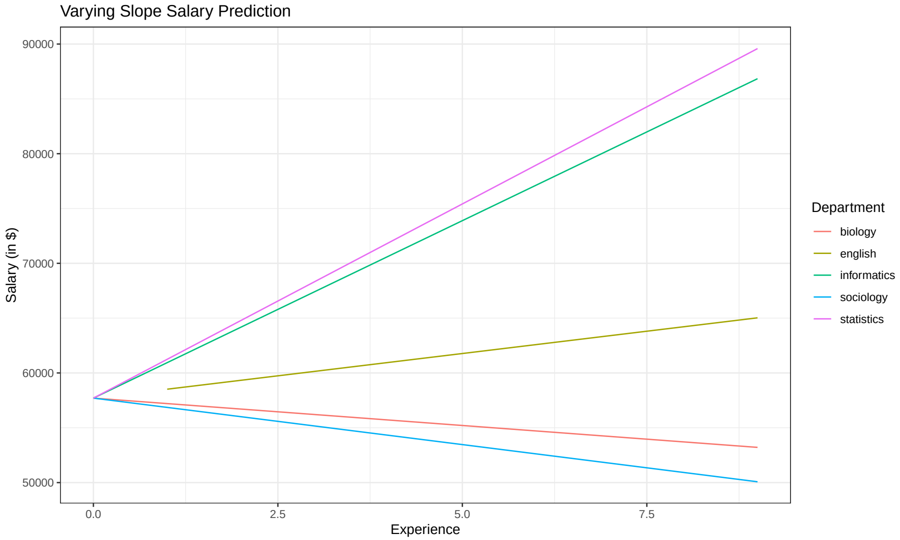
sigma	AIC	BIC	logLik	df.residual
9437.9	2634.86	2646.17	-1313.43	121

Saving the Model m2 predictions

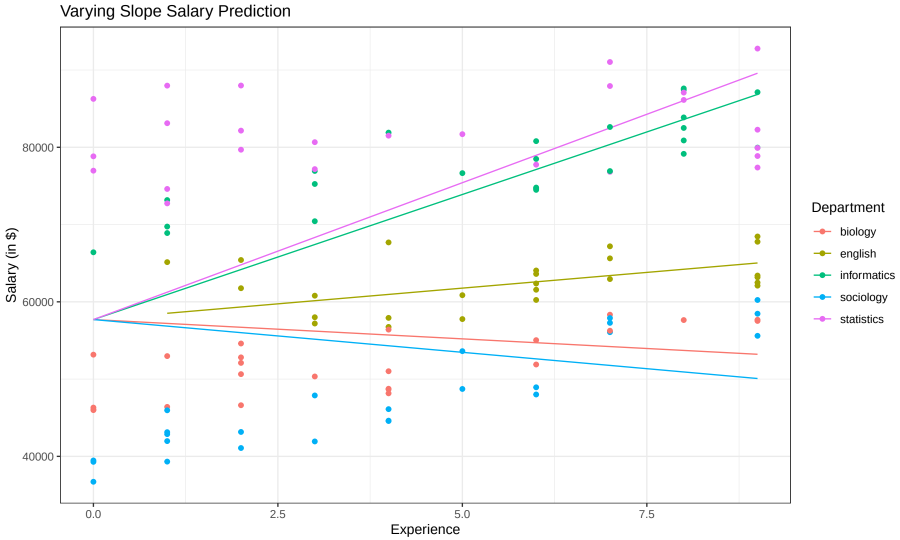
```
facsal$random_slope_preds <- predict(m2)

head(predict(m2))
```

```
       1        2        3        4        5        6 
56009.39 57704.71 65027.81 60941.97 82501.68 50075.79 
```

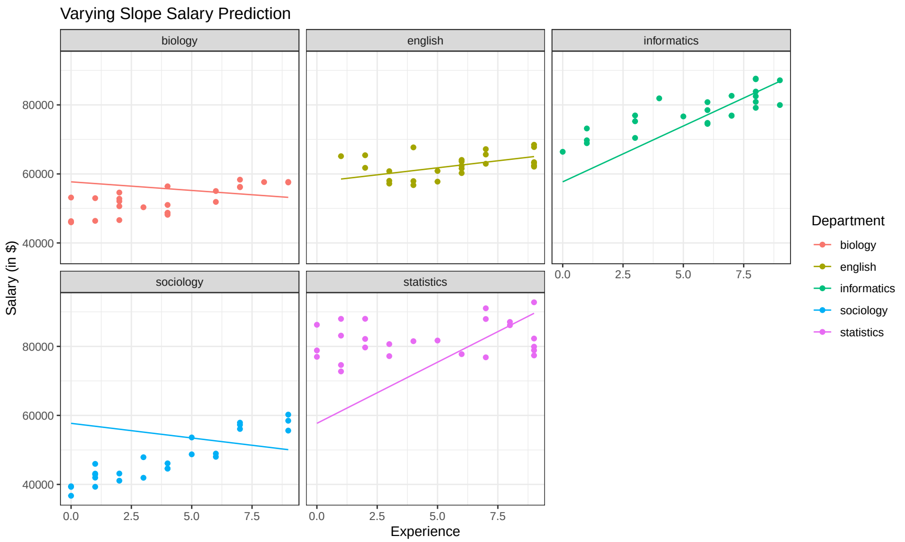
Plotting the `m2` predictions without the data



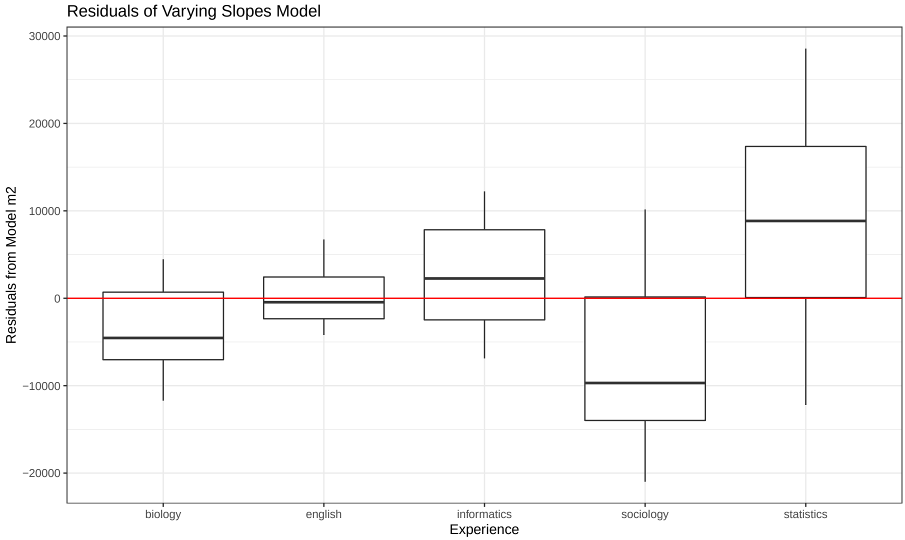
Plotting the `m2` predictions and the data



m2 predictions and the data, faceted by Department



Plot of `m2` Residuals by Department



Let the slopes and intercepts vary

Model with varying slopes and intercept by department

```
m3 <- lmer(salary ~ experience +
               (1 + experience | department),
           data = facsal)
```

Varying Slopes and Intercepts Model

```
m3
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
salary ~ experience + (1 + experience | department)
   Data: facsal
REML criterion at convergence: 2405.105
Random effects:
 Groups     Name        Std.Dev. Corr
 department (Intercept) 16320.1
            experience    722.4  -0.64
 Residual                3569.5
Number of obs: 125, groups:  department, 5
Fixed Effects:
(Intercept)   experience
      59083         1165
```

Tidied m3 Coefficients (use warning = FALSE)

```
tidy(m3) %>%
    kable(digits = 0)
```

effect	group	term	estimate	std.error	statistic
fixed	NA	(Intercept)	59083	7326	8
fixed	NA	experience	1165	342	3
ran_pars	department	sd__(Intercept)	16320	NA	NA
ran_pars	department	cor__(Intercept).experience	-1	NA	NA
ran_pars	department	sd__experience	722	NA	NA
ran_pars	Residual	sd__Observation	3569	NA	NA

Summarizing model `m3`

```
glance(m3) %>%
    select(sigma, AIC, BIC, logLik, df.residual) %>%
    kable(digits = 2)
```

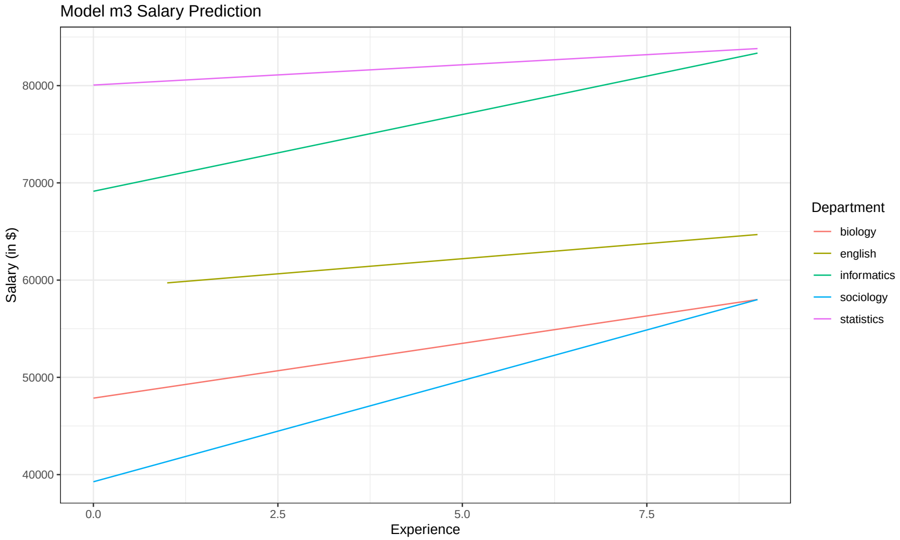
sigma	AIC	BIC	logLik	df.residual
3569.5	2417.11	2434.08	-1202.55	119

Saving the Model `m3` predictions

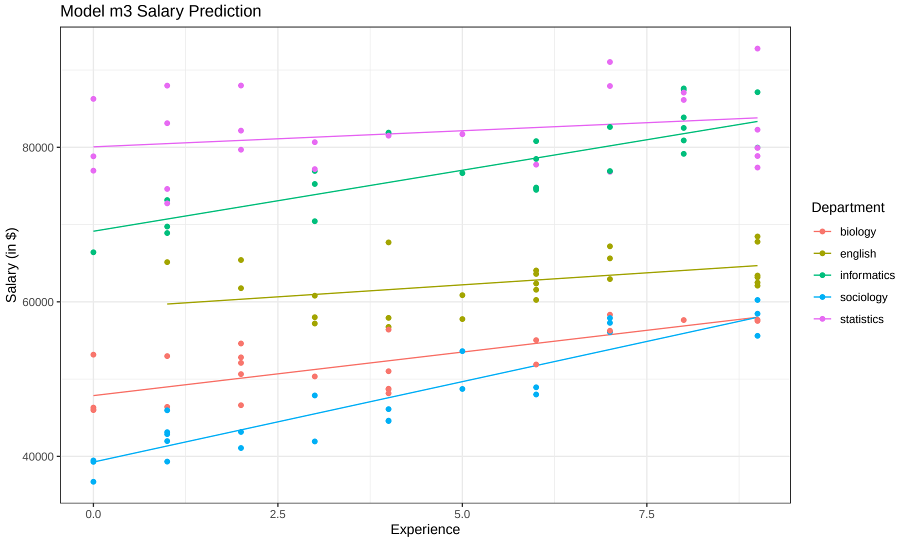
```r
facsal$random_slope_int_preds <- predict(m3)

head(predict(m3))
```

```
       1        2        3        4        5        6 
43426.37 47863.80 64685.23 70714.44 82974.75 57995.15 
```

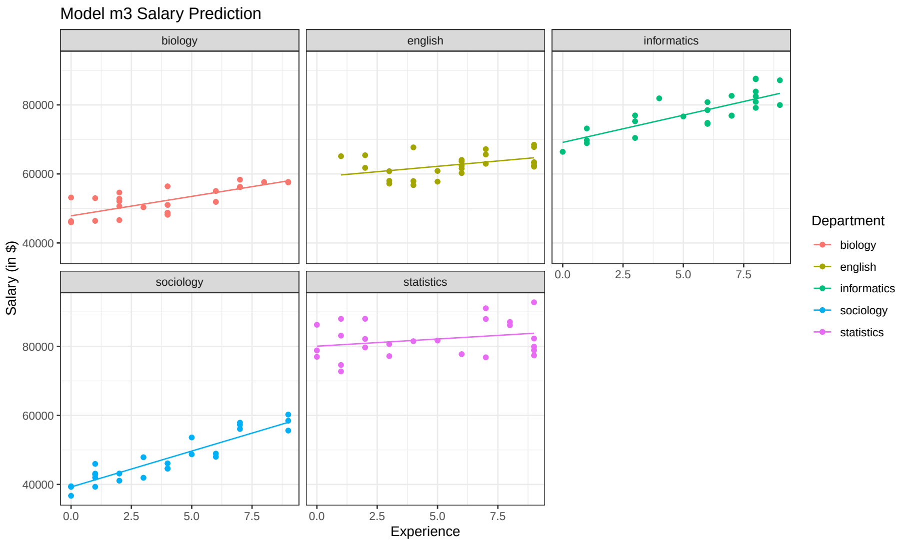
Plotting the m3 predictions without the data



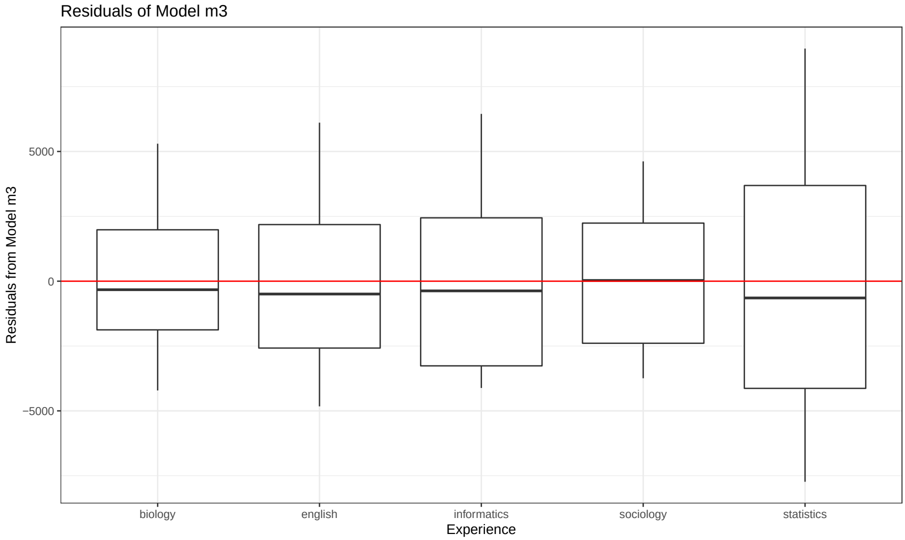
Plotting the `m3` predictions and the data



m3 predictions and the data, faceted by Department



Plot of m3 Residuals by Department



Comparing the Models

```
AIC(m0, m1, m2, m3)
```

```
   df      AIC
m0  3 2741.057
m1  4 2436.544
m2  4 2634.859
m3  6 2417.105
```

```
BIC(m0, m1, m2, m3)
```

```
   df      BIC
m0  3 2749.542
m1  4 2447.857
m2  4 2646.172
m3  6 2434.075
```

Can we test for an effect of experience?

Let's refit model m3 and compare it to an appropriate null model (without the `experience` information), using an `anova` driven likelihood ratio test.

```
m3 <- lmer(salary ~ experience +
               (1 + experience | department),
           data = facsal, REML = FALSE)

m_null <- lmer(salary ~ (1 | department),
               data = facsal, REML = FALSE)
```

The `REML = FALSE` lets us get the likelihood ratio test we want.

Likelihood Ratio Test comparing m3 to m_null

```
anova(m_null, m3)
```

```
Data: facsal
Models:
m_null: salary ~ (1 | department)
m3: salary ~ experience + (1 + experience | department)
       npar    AIC    BIC  logLik deviance  Chisq Df
m_null    3 2527.7 2536.2 -1260.9   2521.7
m3        6 2449.5 2466.5 -1218.8   2437.5 84.196  3
       Pr(>Chisq)
m_null
m3      < 2.2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tidied coefficients from m3

```
tidy(m3, conf.int = TRUE) %>%
    select(-std.error, -statistic) %>%
    kable(digits = 0)
```

effect	group	term	estimate	conf.low	conf.high
fixed	NA	(Intercept)	59078	46212	71944
fixed	NA	experience	1166	566	1766
ran_pars	department	sd__(Intercept)	14610	NA	NA
ran_pars	department	cor__(Intercept).experience	-1	NA	NA
ran_pars	department	sd__experience	636	NA	NA
ran_pars	Residual	sd__Observation	3570	NA	NA

Parametric Bootstrap test for department effect (Part 1)

```
nBoot=100 # should probably be 1000 at a minimum
lrStat=rep(NA,nBoot)
# first fit appropriate null and alternate models
ft.null <- lm(salary ~ experience, data = facsal) #null model
ft.alt <- lmer(salary ~ experience + (1 | department),
               data=facsal, REML=F) # alternate model
# calculate observed test statistic (deviance = -2 * loglik)
lrObs <- 2*logLik(ft.alt) - 2*logLik(ft.null) # test stat
```

Parametric Bootstrap test for department effect (Part 2)

```
set.seed(432)
for(iBoot in 1:nBoot)
{
  facsal$SalSim=unlist(simulate(ft.null)) #resampled data
  # calculate results for our two models in resampled data
  bNull <- lm(SalSim ~ experience,
              data=facsal) #null model
  bAlt <- lmer(SalSim ~ experience + (1|department),
               data=facsal, REML=F) # alternate model
  # calculate and store resampled test stat
  lrStat[iBoot] <- 2*logLik(bAlt) - 2*logLik(bNull)
}
```

```
boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
```

Parametric Bootstrap Test for Department effect (Part 3)

```r
mean(lrStat>lrObs) # P-value for test of department effect
```

```
[1] 0
```

Even this "simple" model isn't simple.

Our parametric bootstrap repeatedly hits up on the edge of a problem with the random effects.

`boundary (singular) fit: see ?isSingular`

is the warning we've received above.

What is a Mixed Model?

A model for an outcome that incorporates both fixed and random effects.

Or, alternatively,...

> *Mixed models are those with a mixture of fixed and random effects. Random effects are categorical factors where the levels have been selected from many possible levels and the investigator would like to make inferences beyond just the levels chosen.*

- From http://environmentalcomputing.net/mixed-models/

A Random Effect?

A random factor:

- is categorical
- has a large number of levels
- only a subsample (often a random subsample) of levels is included in your design
- you want to make inference in general, and not only for the levels you observed

Think of a random factor as a group where:

- you want to quantify variation between group levels
- you want to make predictions about unobserved groups
- but you don't want to compare outcome differences between particular group levels

Sources: https://bbolker.github.io/morelia_2018/notes/glmm.html and http://environmentalcomputing.net/mixed-models-1/

Why Use a Random Effect?

- You want to combine information across groups
- You have variation in information per group level (number of samples or amount of noisiness)
- You have a categorical predictor that is a nuisance variable (something not of direct interest but that we want to control for)
- You have more than 5-6 groups

Source: Crawley (2002) and Gelman (2005) quoted at https://bbolker.github.io/morelia_2018/notes/glmm.html

What is a Fixed Effect vs. a Random Effect?

The one I most often use is something like:

- Fixed effects are constant across individuals, while random effects vary.

The various definitions in the literature are incompatible with each other[1].

From Scahabenberger and Pierce (2001), we have this gem:

> *One modeler's random effect is another modeler's fixed effect.*

A more practical definition might be to ask the question posed by Crawley (2002):

> *Are there enough levels of the factor in the data on which to base an estimate of the variance of the population of effects? No, means [you should probably treat the variable as] fixed effects.*

[1] See, for instance, the GLMM FAQ referenced earlier

Models We Might Consider

Suppose we have an outcome `y`, predictor `x` and group `group`

- `y ~ x` = linear regression on `x`: not a mixed model
- `y ~ 1 + (1 | group)` = random intercept on group: null model
- `y ~ x + (1 | group)` = fixed slope and random intercept
- `y ~ (0 + x | group)` = random slope of x within group, no variation in intercept
- `y ~ x + (x | group)` = random intercept and random slope

A "More" Realistic Example

The most common example in modern medicine has measurements nested within people. Repeated measures and longitudinal data provide typical settings for this sort of approach.

Another setting where a hierarchical approach is of interest occurs when you have variables measured at multiple levels, for instance you have information on patients, who are nested within providers, who are nested within hospitals.

Nothing of what I've talked about today should be taken as the final word on how to extend these ideas beyond the very simple example I've provided this afternoon.