Mixed effects models

Wheels within wheels

Samuel Robinson, Ph.D.

December 10, 2020

Motivation

- What are mixed effects models?
  - Scary math (matrix algebra)
  - Variance partitioning
  - Fixed effects vs. Random effects
- Working with random effects
  - What should be a random effect?
  - Model validation
  - Hypothesis testing
  - Examples
- Exercise!

What are mixed effects models?

Many different names:

1. Mixed effects models
2. Random effects models
3. Heirarchical models
4. Empirical/Bayesian heirarchical models
5. Latent variable models
6. Split-plot models[1]

I will use the term *heirarchical models*, as this is the closest to what I will teach you

[1] Earlier form of variance partitioning

Scary math

Unfortunately, we need a review of matrix algebra in order to explain this:

- This is a matrix:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

- This is a vector

$$b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

- Multiplying them looks like this:

$$A \times b = Ab = 1 \times \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 2 \times \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} + 3 \times \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 30 \\ 36 \\ 42 \end{bmatrix}$$

Why do we call them "linear models"?

Involves a *linear mapping* of coefficients onto a model matrix

Coefficients:

$$\beta = \begin{bmatrix} 0.1 & 1.8 & -0.03 \end{bmatrix}$$

Model matrix:

$$X = \begin{bmatrix} 1 & 1 & 10 \\ 1 & 1 & 12 \\ 1 & 0 & 9 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Multiplying them looks like:

$$\hat{y} = X\beta = \begin{bmatrix} 1.60 \\ 1.54 \\ -0.17 \\ \vdots \end{bmatrix}$$

This is exactly what R does to fit models:

```
head(dat)
```

```
##            y         x group
## 1 -3.9768978 -4.248450     c
## 2 -8.1170395  5.766103     h
## 3 -1.7686615 -1.820462     a
## 4  2.0112311  7.660348     g
## 5 -0.8777605  8.809346     g
## 6 -2.0307915 -9.088870     g
```

```
m1 <- lm(y~x,data=dat)
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7923  -3.7808  -0.5984   3.4144  14.0325
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.12032    0.43464   4.878 2.73e-06 ***
## x           -0.22430    0.07593  -2.954  0.00365 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.323 on 148 degrees of freedom
## Multiple R-squared:  0.05569,    Adjusted R-squared:  0.04931
## F-statistic: 8.728 on 1 and 148 DF,  p-value: 0.003647
```

This is exactly what R does to fit models (cont.):

```r
head(model.matrix(m1))
```

```
##   (Intercept)         x
## 1           1 -4.248450
## 2           1  5.766103
## 3           1 -1.820462
## 4           1  7.660348
## 5           1  8.809346
## 6           1 -9.088870
```

```r
coef(m1)
```

```
## (Intercept)           x
##   2.1203243  -0.2243028
```

```r
pred2 <- model.matrix(m1) %*% coef(m1) #predicted = matrix * coefs
head(data.frame(pred1=predict(m1),pred2)) #same thing!
```

```
##       pred1     pred2
## 1 3.0732633 3.0732633
## 2 0.8269716 0.8269716
## 3 2.5286589 2.5286589
## 4 0.4020872 0.4020872
## 5 0.1443638 0.1443638
## 6 4.1589829 4.1589829
```

Groups are coded by "dummy variables" (0s and 1s)

```
head(dat)
```

```
##            y         x group
## 1 -3.9768978 -4.248450     c
## 2 -8.1170395  5.766103     h
## 3 -1.7686615 -1.820462     a
## 4  2.0112311  7.660348     g
## 5 -0.8777605  8.809346     g
## 6 -2.0307915 -9.088870     g
```

```
m2 <- lm(y~group,data=dat)
head(model.matrix(m2)) #0s and 1s used to identify groups
```

```
##   (Intercept) groupb groupc groupd groupe groupf groupg grouph groupi groupj
## 1           1      0      1      0      0      0      0      0      0      0
## 2           1      0      0      0      0      0      0      1      0      0
## 3           1      0      0      0      0      0      0      0      0      0
## 4           1      0      0      0      0      0      1      0      0      0
## 5           1      0      0      0      0      0      1      0      0      0
## 6           1      0      0      0      0      0      1      0      0      0
```

```
coef(m2) #This uses the 1st group as the "control" group
```

```
## (Intercept)      groupb      groupc      groupd      groupe      groupf
##   2.6272010   1.8996085  -3.1245414  -0.5774078  -3.0486394  -1.1922202
##      groupg      grouph      groupi      groupj
##  -4.1358117  -7.8251986   8.4839175   2.0746211
```

Structure of LMs... now with matrices!

- All linear models take the form:

$$\hat{y} = X\beta = b_0 1 + b_1 x_1 ... + b_i x_i$$
$$y \sim Normal(\hat{y}, \sigma)$$

- y is a vector of data you want to predict
- $\hat{y}$ is a vector of *predicted values* for y
- $X = \{1, x_1 ...\}$ is a matrix of *predictors* for y
- $\beta = \{b_0, b_1, ...\}$ is a vector of *coefficients*
- $y \sim Normal(\hat{y}, \sigma)$ means:
  - "y follows a Normal distribution with mean $\hat{y}$ and SD σ"

Fixed effects vs. Random effects

Say that X is a model matrix coding for 10 sites[2], and y is something we're interested in predicting

$$\hat{y} = b_0 + X\beta$$
$$y \sim Normal(\hat{y}, \sigma)$$

- Site coefficients (β) are unrelated to each other
- σ is the SD of *residuals*
- Site is a **fixed effect**

$$\hat{y} = b_0 + X\beta$$
$$y \sim Normal(\hat{y}, \sigma)$$
$$\beta \sim Normal(0, \sigma_{site})$$

- Site coefficients (β) are related to each other via a *Normal* distribution
- σ is the SD of *residuals*, σ_{site} is the SD of *sites*
- Site is a **random effect**

[2] Intercept is a separate variable