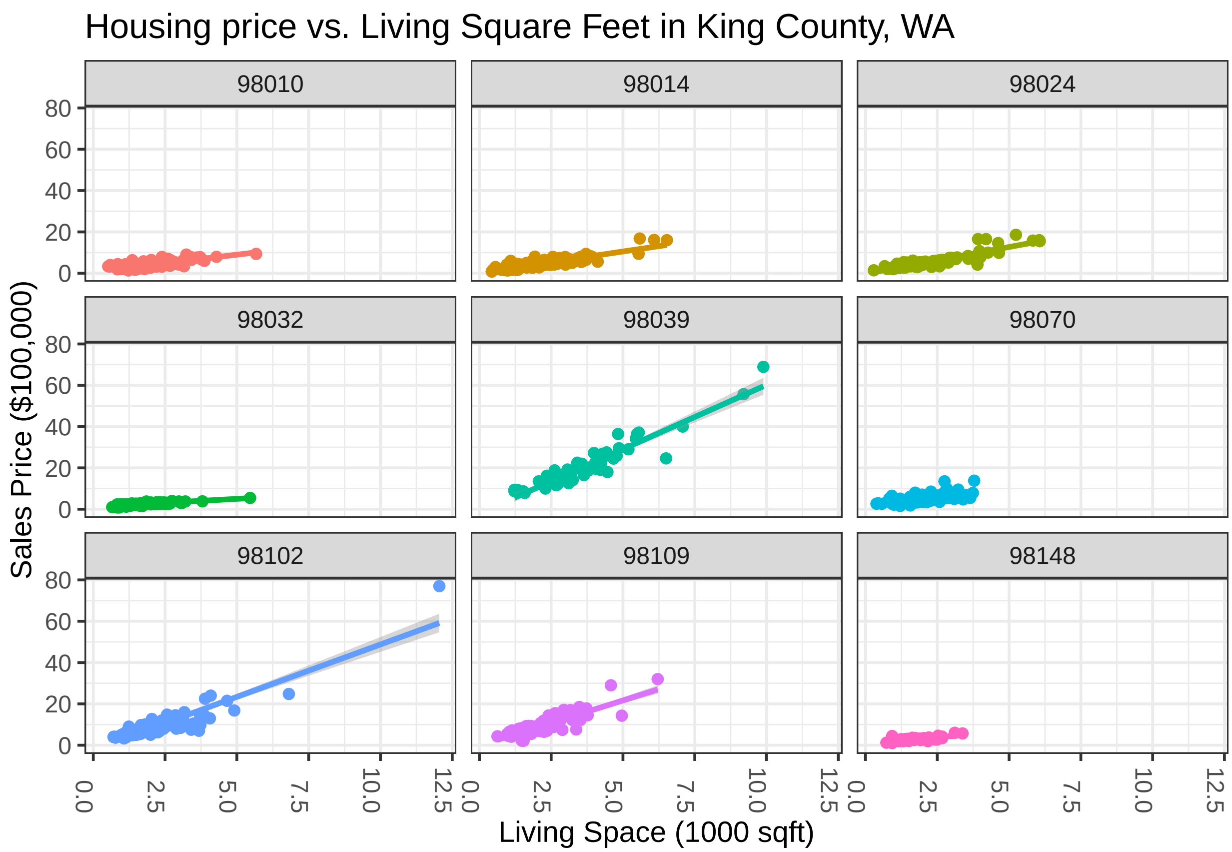
Hierarchical Models

Motivating Dataset

Recall the housing dataset from King County, WA that contains sales prices of homes across the Seattle area. Below we see the relationship between sales price and the size of the home across several zipcodes.



Multilevel models

While we will initially just look at a model with the varying intercepts, this approach can also be applied to covariates.

There are several different, but equivalent specifications in GH 12.5, but here is one way to look at the model.

$$\begin{aligned} y_i &\sim N(\alpha_{j[i]} + X_i \underline{\beta}, \sigma_y^2) \\ \alpha_j &\sim N(\mu_\alpha, \sigma_\alpha^2) \end{aligned}$$

lmer

One common approach for hierarchical models is to use the `lmer` function in the `lme4` package. Note that the hierarchical structure we have detailed can also be applied to GLMs using `glmer`. Note that most of this code (and the textbook) is "pre-rstanarm", so it might be more intuitive to use `stan_glmer`, which we will also look at a Bayesian version in a little bit using `stan_glmer`.

We need to denote what terms will vary by group.

```
lmer1 <- lmer(price ~ (1 | zipcode) , data = seattle)
display(lmer1)
```

```
## lmer(formula = price ~ (1 | zipcode), data = seattle)
##  coef.est   coef.se
## 713204.24 195580.94
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 584666.88
##  Residual             460890.53
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 25155.1, DIC = 25201.4
## deviance = 25175.2
```

```
coef(lmer1)
```

```
## $zipcode
##       (Intercept)
## 98010    425454.1
## 98014    456901.5
## 98024    581647.2
## 98032    253581.2
## 98039   2143523.7
## 98070    488663.0
## 98102    900408.3
## 98109    879131.8
## 98148    289527.5
##
## attr(,"class")
## [1] "coef.mer"
```

Note the coefficients for a specific group are defined as the fixed effect + the random effect.

```
fixef(lmer1)
```

```
## (Intercept)
##    713204.2
```

The fixed effect here corresponds to μ_α. The standard component associated with the random effect can also be extracted.

```
sigma.hat(lmer1)$sigma$zipcode
```

```
## (Intercept)
##    584666.9
```

The takeaway idea is that the zipcode level intercept (mean) comes from a distribution.

$$\alpha_j \sim N(713,204;\quad 584,666.9^2)$$

The estimated "random effects" are often decomposed as the sum of μ_α and zero-centered random deviations from $N(0, \sigma^2_\alpha)$.

```
ranef(lmer1)
```

```
## $zipcode
##       (Intercept)
## 98010   -287750.1
## 98014   -256302.7
## 98024   -131557.1
## 98032   -459623.1
## 98039   1430319.4
## 98070   -224541.3
## 98102    187204.0
## 98109    165927.6
## 98148   -423676.7
##
## with conditional variances for "zipcode"
```

```
fixed_ci <- round(fixef(lmer1)['(Intercept)'] + c(-2,2) * se.fixef(lmer1)['(Intercept)'])
```

Summarizing the model The 95% confidence interval for the fixed effects intercept is (322,042, 1,104,366). This can be interpreted as the overall mean price of a house. Formally, this is more the mean of the group means.

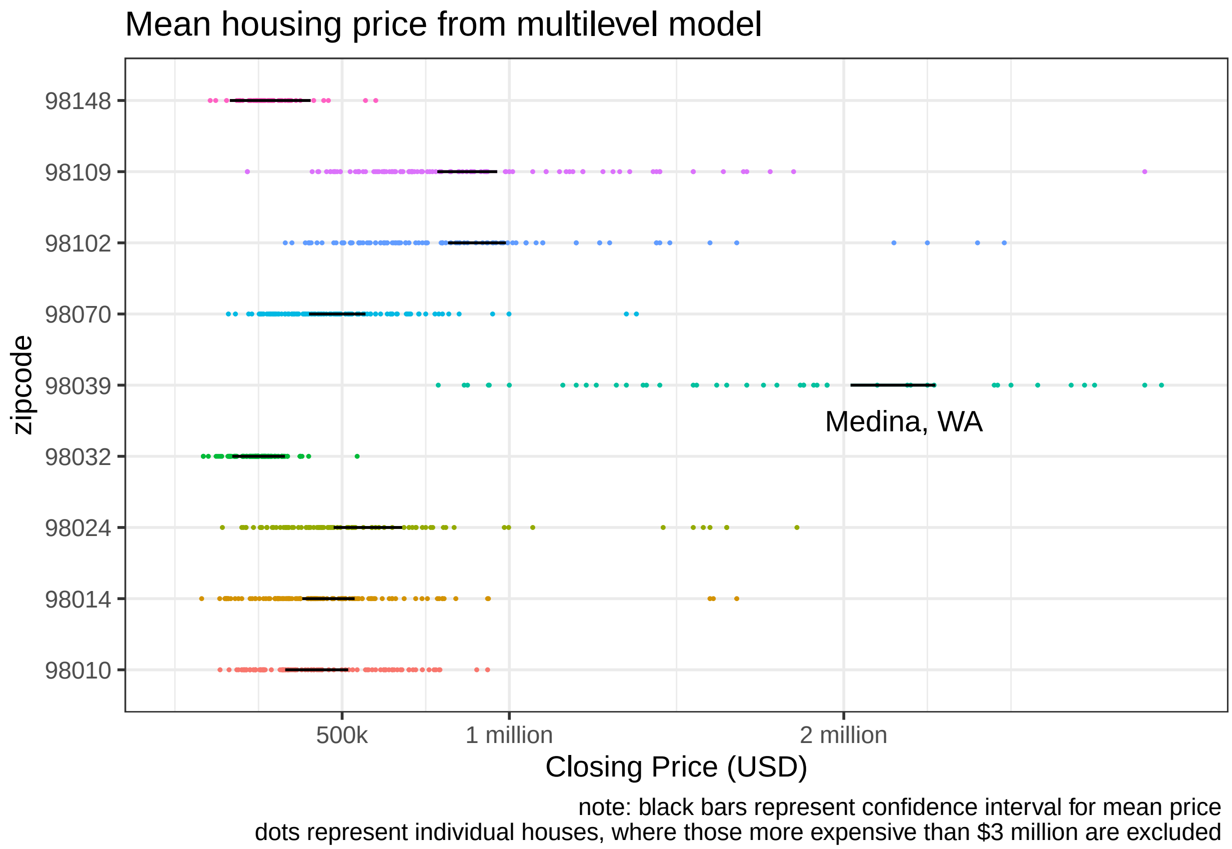
The 95% confidence intervals for the group effects (or deviations from the mean price) are:

zipcode	lower	upper
98010	-379643	-195857
98014	-338874	-173731
98024	-233587	-29528
98032	-541866	-377381
98039	1300763	1559876
98070	-309176	-139907
98102	97512	276896
98109	77888	253968
98148	-545109	-302244

A more useful way to summarize the data would be to create 95% confidence intervals for the overall intercept (fixed effect + random effect) for each group. In other words, we are now asking what are the plausible range of values for prices in each zipcode. To answer this question, we can use the `sim` function.

```
samples <- arm::sim(lmer1, n.sims = 1000)
overall <- fixef(samples)
group <- matrix(ranef(samples)$zipcode[,,1], nrow = 1000, ncol = ngrps(lmer1), byrow = F)
group_totals <- group + matrix(overall, nrow = 1000, ncol = ngrps(lmer1))
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



Prediction

Note the previous figure contains uncertainty for the mean price within a particular zipcode. Similar to before you could also make predictions for a new home in an existing dataset.

Predictions can also be made for a new zipcode. This requires drawing a group level effect from the hierarchical distribution for group effects.

```
sigma_alpha <- sigma.hat(lmer1)$sigma$zipcode
mu_alpha <- fixef(lmer1)["(Intercept)"]
rnorm(10, mu_alpha, sigma_alpha)
```

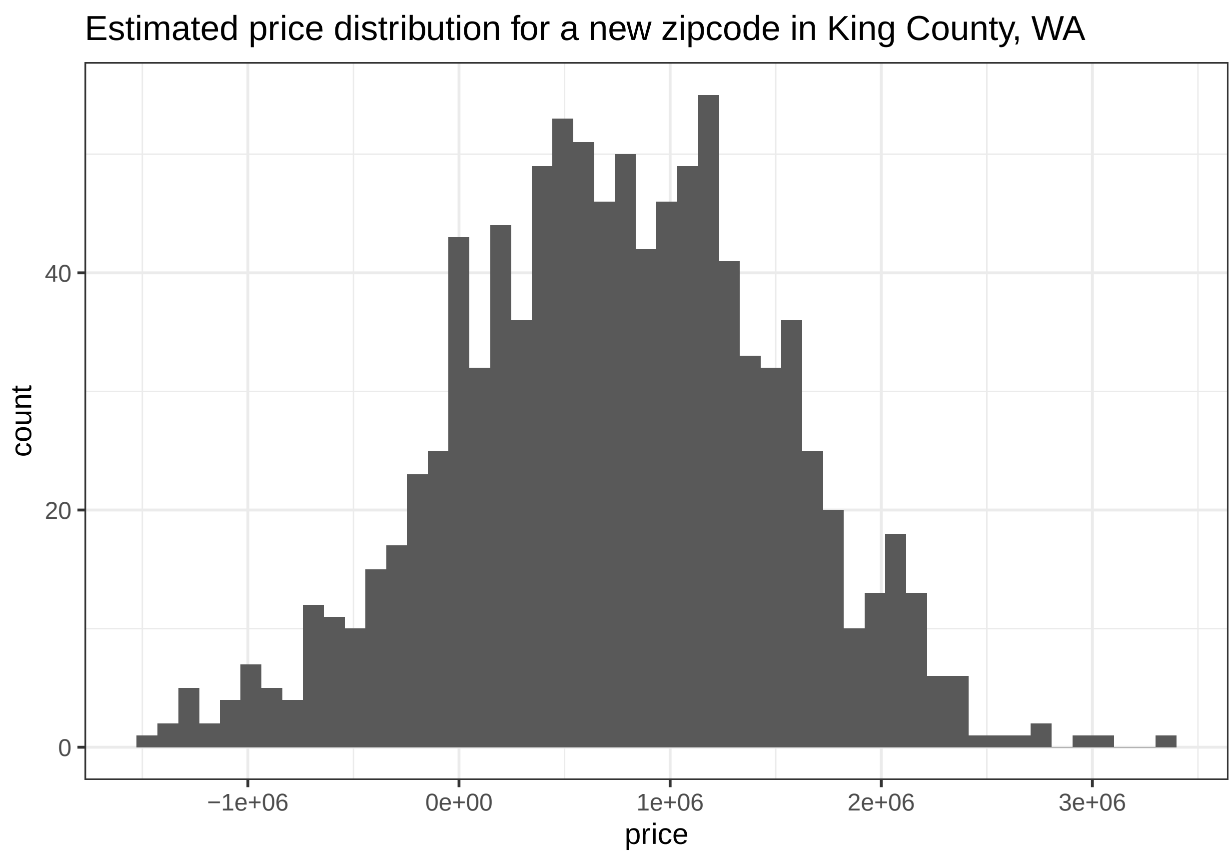
```
##  [1] -135431.0  246302.8 -213755.5  137600.7  614944.2  714019.6 1085055.8
##  [8] -419585.9  206712.7 1113341.9
```

```
alpha_samples <- rnorm(1000, mu_alpha, sigma_alpha)
```

Then using each of those sampled random effects to draw an individual response (with the appropriate data level variance).

```
sigma_y <- sigma.hat(lmer1)$sigma$data

new_zip <- rnorm(1000, mean = alpha_samples, sd = sigma_y)
```



Adding Coefficients The model we have just outlined does not include any additional covariates.

- First, consider a single covariate with the same effect across all of the groups. This is often referred to as a random-intercept, fixed-slope model.

```
lmer2 <- lmer(price ~ scale_sqft + (1 |zipcode), data = seattle)

display(lmer2)
```

```
## lmer(formula = price ~ scale_sqft + (1 | zipcode), data = seattle)
##             coef.est  coef.se
## (Intercept) 682210.16 127976.83
## scale_sqft  403385.07  10167.55
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 382797.06
##  Residual             274619.10
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 24238.4, DIC = 24321.6
## deviance = 24276.0
```

- Next, we can also consider a covariate that varies across groups. This corresponds to a varying-slope and varying-intercept model. It is also possible to have a varying-slope model with a fixed intercept.

$$
\begin{aligned}
y_i &\sim N(\alpha_{j[i]} + X_i\beta_{j[i]}, \sigma_y^2) \\
\alpha_j &\sim N(\mu_\alpha, \sigma_\alpha^2) \\
\beta_j &\sim N(\mu_\beta, \sigma_\beta^2)
\end{aligned}
$$

Note: you may have to adjust the REML and optimizer options to achieve convergence

```
lmer_nonconverge <- lmer(price ~ scale_sqft + (1 + scale_sqft|zipcode), data = seattle)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00246273 (tol = 0.002, component 1)
```

```
lmer3 <- lmer(price ~ scale_sqft + (1 + scale_sqft|zipcode), data = seattle,
      REML = FALSE)
display(lmer3)
```

```
## lmer(formula = price ~ scale_sqft + (1 + scale_sqft | zipcode),
##     data = seattle, REML = FALSE)
##             coef.est  coef.se
## (Intercept) 606247.84  90723.36
## scale_sqft  330602.83  69904.63
##
## Error terms:
##  Groups   Name        Std.Dev.  Corr
##  zipcode  (Intercept) 271254.68
##           scale_sqft  208120.29 0.99
##  Residual             196377.48
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 23716.8, DIC = 23704.8
## deviance = 23704.8
```

The fixed-effects or means of the group-level effects can be extracted.

```
fixef(lmer3)
```

```
## (Intercept)  scale_sqft
##    606247.8    330602.8
```

Similarly, the variance of those group-level effects can also be obtained from the model.

```
sigma.hat(lmer3)$sigma
```

```
## $data
## [1] 196377.5
##
## $zipcode
## (Intercept)  scale_sqft
##    271254.7    208120.3
```

stan_glmer

Similar to how we have used `stan_glm()`, we can also use `stan_glmer()` to fit these models.

```
stan_lmer1 <- stan_glmer(price ~ (1 | zipcode) , data = seattle)
```

```
print(stan_lmer1)
```

```
## stan_glmer
##  family:       gaussian [identity]
##  formula:      price ~ (1 | zipcode)
##  observations: 869
## ------
##             Median   MAD_SD
## (Intercept) 720051.6 186543.3
##
## Auxiliary parameter(s):
##       Median   MAD_SD
## sigma 461411.2  11379.4
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 647744
##  Residual             461387
## Num. levels: zipcode 9
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
display(lmer1)
```

```
## lmer(formula = price ~ (1 | zipcode), data = seattle)
##  coef.est   coef.se
## 713204.24 195580.94
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 584666.88
##  Residual             460890.53
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 25155.1, DIC = 25201.4
## deviance = 25175.2
```

```r
coef(stan_lmer1)
```

```
## $zipcode
##       (Intercept)
## 98010    433620.9
## 98014    459609.4
## 98024    582635.4
## 98032    259477.4
## 98039   2142106.6
## 98070    491242.3
## 98102    901322.4
## 98109    881198.8
## 98148    287707.1
##
## attr(,"class")
## [1] "coef.mer"
```

```r
coef(lmer1)
```

```
## $zipcode
##       (Intercept)
## 98010    425454.1
## 98014    456901.5
## 98024    581647.2
## 98032    253581.2
## 98039   2143523.7
## 98070    488663.0
## 98102    900408.3
## 98109    879131.8
## 98148    289527.5
##
## attr(,"class")
## [1] "coef.mer"
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       gaussian [identity]
##  formula:      price ~ (1 | zipcode)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 869
##  groups:       zipcode (9)
##
## Estimates:
##                                         mean          sd
## (Intercept)                            7.180202e+05  2.072531e+05
## b[(Intercept) zipcode:98010]          -2.922107e+05  2.116191e+05
## b[(Intercept) zipcode:98014]          -2.613705e+05  2.104071e+05
## b[(Intercept) zipcode:98024]          -1.380639e+05  2.125176e+05
## b[(Intercept) zipcode:98032]          -4.638964e+05  2.098118e+05
## b[(Intercept) zipcode:98039]           1.424663e+06  2.161721e+05
## b[(Intercept) zipcode:98070]          -2.283068e+05  2.102849e+05
## b[(Intercept) zipcode:98102]           1.820903e+05  2.111563e+05
## b[(Intercept) zipcode:98109]           1.607579e+05  2.105069e+05
## b[(Intercept) zipcode:98148]          -4.292069e+05  2.136176e+05
## sigma                                  4.613871e+05  1.128380e+04
## Sigma[zipcode:(Intercept),(Intercept)]  4.195725e+11  2.366775e+11
##                                         2.5%          97.5%
## (Intercept)                            3.139399e+05  1.140375e+06
## b[(Intercept) zipcode:98010]          -7.337727e+05  1.156100e+05
## b[(Intercept) zipcode:98014]          -6.918456e+05  1.417519e+05
## b[(Intercept) zipcode:98024]          -5.677598e+05  2.755605e+05
## b[(Intercept) zipcode:98032]          -8.905088e+05 -5.647870e+04
## b[(Intercept) zipcode:98039]           9.902394e+05  1.847178e+06
## b[(Intercept) zipcode:98070]          -6.586393e+05  1.757602e+05
## b[(Intercept) zipcode:98102]          -2.453542e+05  5.984881e+05
## b[(Intercept) zipcode:98109]          -2.676469e+05  5.728120e+05
## b[(Intercept) zipcode:98148]          -8.677876e+05 -7.893300e+03
## sigma                                  4.395936e+05  4.837276e+05
## Sigma[zipcode:(Intercept),(Intercept)]  1.581070e+11  1.062655e+12
##
## Fit Diagnostics:
##           mean     sd       2.5%     97.5%
## mean_PPD 632532.1  22491.1 589519.4 676291.2
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                        mcse         Rhat         n_eff
## (Intercept)                                  8096.0          1.0  655
## b[(Intercept) zipcode:98010]                 8093.0          1.0  684
## b[(Intercept) zipcode:98014]                 8081.9          1.0  678
## b[(Intercept) zipcode:98024]                 8171.3          1.0  676
## b[(Intercept) zipcode:98032]                 8037.1          1.0  681
## b[(Intercept) zipcode:98039]                 8073.7          1.0  717
## b[(Intercept) zipcode:98070]                 8072.3          1.0  679
```

```
## b[(Intercept) zipcode:98102]                8125.3          1.0  675
## b[(Intercept) zipcode:98109]                8072.3          1.0  680
## b[(Intercept) zipcode:98148]                8053.8          1.0  704
## sigma                                        262.5          1.0 1847
## Sigma[zipcode:(Intercept),(Intercept)] 8057491369.9          1.0  863
## mean_PPD                                     361.9          1.0 3861
## log-posterior                                  0.1          1.0  747
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

We can also directly extract the simulations from the stan object.

```
##           parameters
## iterations (Intercept) b[(Intercept) zipcode:98010]
##       [1,]    553065.6                  -175098.576
##       [2,]    440951.9                     5120.821
##       [3,]    550855.2                  -159650.767
##       [4,]    705690.3                  -223389.274
##       [5,]    673511.5                  -313228.718
##       [6,]    687668.8                  -243363.529
##           parameters
## iterations b[(Intercept) zipcode:98014] b[(Intercept) zipcode:98024]
##       [1,]                   -122247.94                    121698.68
##       [2,]                    -10049.28                     82638.26
##       [3,]                    -84161.51                     51599.79
##       [4,]                   -267511.71                    -86230.00
##       [5,]                   -190006.36                   -138915.65
##       [6,]                   -311298.08                    -70309.25
##           parameters
## iterations b[(Intercept) zipcode:98032] b[(Intercept) zipcode:98039]
##       [1,]                    -316417.5                      1609469
##       [2,]                    -228110.1                      1663590
##       [3,]                    -296829.6                      1727883
##       [4,]                    -455534.7                      1340954
##       [5,]                    -419477.2                      1422103
##       [6,]                    -469326.7                      1484006
##           parameters
## iterations b[(Intercept) zipcode:98070] b[(Intercept) zipcode:98102]
##       [1,]                   -74833.627                     272358.7
##       [2,]                    -3933.769                     474010.5
##       [3,]                   -37949.571                     337630.0
##       [4,]                  -249930.820                     226629.5
##       [5,]                  -160712.388                     170048.4
##       [6,]                  -253185.070                     224196.6
##           parameters
## iterations b[(Intercept) zipcode:98109] b[(Intercept) zipcode:98148]    sigma
##       [1,]                     265694.0                    -306732.9 487358.7
##       [2,]                     467718.4                    -182148.1 479089.8
##       [3,]                     288852.8                    -230409.2 448435.8
##       [4,]                     213821.6                    -422659.4 457941.7
##       [5,]                     161492.4                    -380825.0 457070.6
##       [6,]                     170426.0                    -420126.7 448662.7
##           parameters
## iterations Sigma[zipcode:(Intercept),(Intercept)]
##       [1,]                           228980805686
##       [2,]                           253110054490
##       [3,]                           413825607264
##       [4,]                           811398657455
##       [5,]                           835295036631
##       [6,]                           809567836989
```

This can be used for generating predictions and credible intervals.

Final Connections **Group-level covariates**: consider modeling test scores by school district. There would be school district level covariates that could be important - such as percent of free and reduced lunch. These type of variables could be incorporated as group-level covariates.

$$\begin{aligned} y_i &\sim N(\alpha_{j[i]} + X_i \beta_{j[i]}, \sigma_y^2) \\ \alpha_j &\sim N(\mu_\alpha, \sigma_\alpha^2) \\ \beta_j &\sim N(\underline{\boldsymbol{u_j} \times \boldsymbol{\mu_\beta}}, \sigma_\beta^2) \end{aligned}$$

where u_j is a group-level covariate.

Interactions: recall that interactions provide a way for different relationships of a response across a set of group. Multilevel models provide a natural way to do this *and* provide benefits of shrinkage.

Shrinkage: Recall the estimate value for a group is a weighted average from the data in that group and the overall data.

$$\hat{\alpha}_j \approx \frac{\frac{n_j}{\sigma_y^2}\bar{y}_j + \frac{1}{\sigma_\alpha^2}\bar{y}_{all}}{\frac{n_j}{\sigma_y^2} + \frac{1}{\sigma_\alpha^2}}$$

where σ_y^2 is the variance of the data and σ_α^2 is the variance of the group-level averages. So, in the limits, large data variance (relative to the group variances) converges to complete pooling and large group variance (relative to the data variance) converges to the individual group means.

Selection of Random Effects: these varying effect models necessarily impose additional complexity on our modeling framework; however, GH suggest embracing the complexity (as it often helps directly answer research questions), moreover, they don't recommend using evidence statements to select specific random effects.