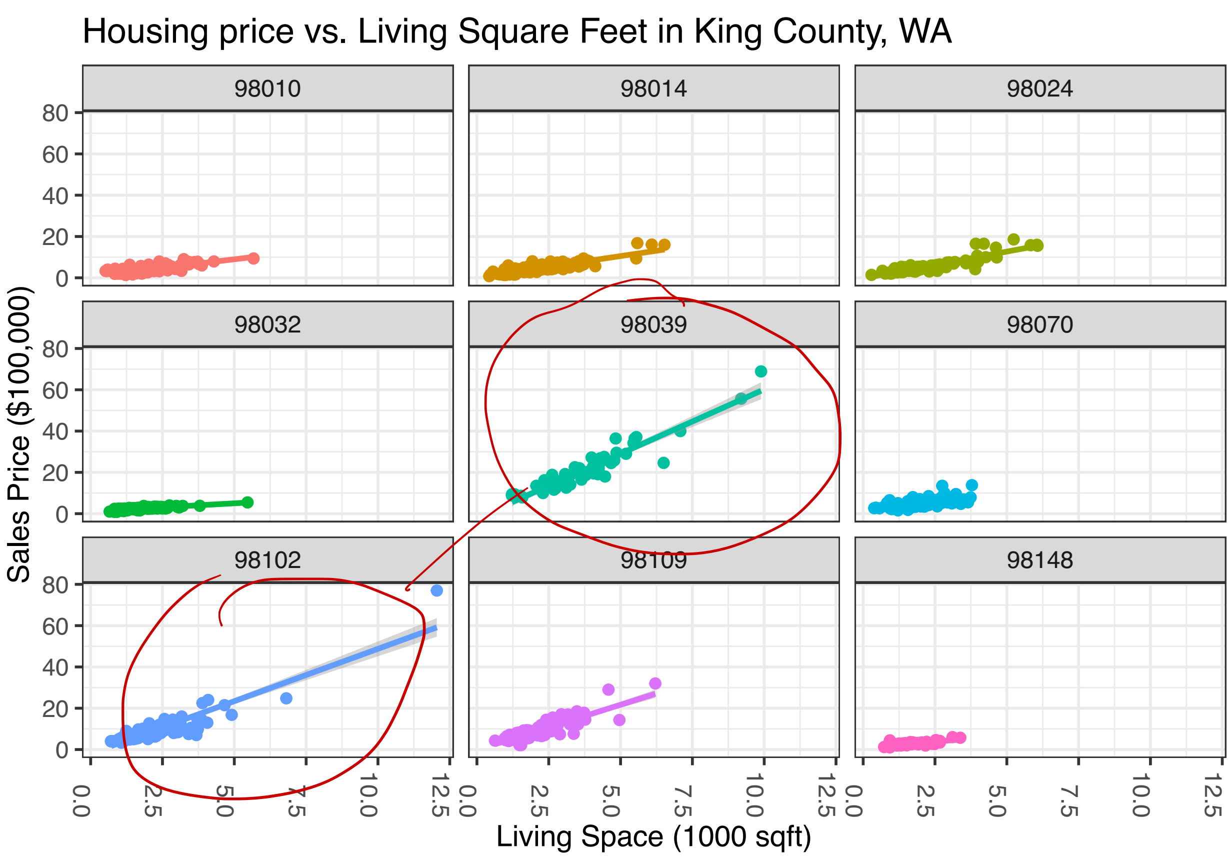
Lecture 15: Gelman Hill Ch 12 + Ch 13

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.



$X_i = A(\ldots$

$\cancel{\alpha}$

Multilevel models

$$y_i \sim N\left(\alpha_{j[i]} + X_i \beta_{j[i]},\ \sigma_y^2\right)$$

$$\alpha_j \sim N\left(\mu_\alpha,\ \sigma_\alpha^2\right)$$

$$\beta_j \sim N\left(\mu_\beta,\ \sigma_\beta^2\right)$$

lmer

One common approach for hierarchical models is to use the `lmer` function in the $\boxed{\texttt{lme4}}$ package. Note that the hierarchical structure we have detailed can also be applied to GLMs using `glmer`.

intercept will vary be zipcode

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 195580.98
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 584667.00
##  Residual             460890.53
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 25155.1, DIC = 25201.4
## deviance = 25175.2
```

μ_α σ_α

σ_α σ_y

Q: where are:

- α_j
- σ^2_y
- σ^2_α

```
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"
```

α_j these are weighted averages

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

```
fixef(lmer1)
```

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

α

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)
##      584667
```

σ_α

$$\alpha_j \sim N(713{,}000,\ 584{,}000^2)$$

random effects can be thought of as deviations from the overall mean

```r
ranef(lmer1)
```

$\hat{\theta}$

```
## $zipcode
##         (Intercept)
## 98010   -287750.1
## 98014   -256302.7
## 98024   -131557.1
## 98032   -459623.1
## 98039   1430319.5
## 98070   -224541.3
## 98102    187204.0
## 98109    165927.6
## 98148   -423676.7
##
## with conditional variances for "zipcode"
```

$\alpha_j - \mu_\alpha \sim N(0, \sigma_\alpha^2)$

(baseline)

looks like the reference case formulation

Summarizing the model

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

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.

μ_α

Sample from these

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	-545110	-302244

function in a specific package

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.

draw μ_α

```r
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).
```

note: black bars represent confidence interval for mean price, dots represent individual houses, where those more expensive than $3 million are excluded

α_j from a hierarchical framework → shrinkage

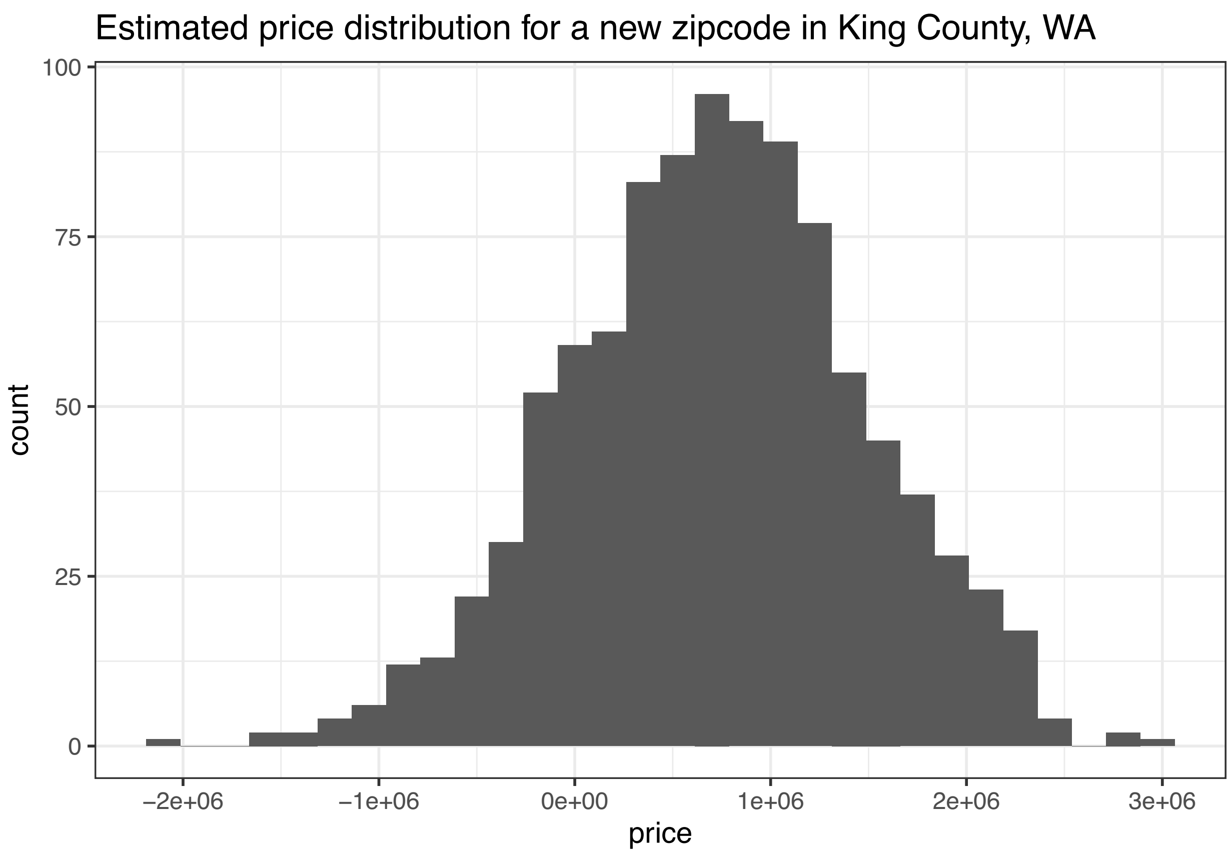
Prediction

the figure above is about mean prices,

Q: how to make predictions for an individual house in an existing zipcode?

1. sample μ_α
2. sample deviation from μ_α for group j
3. We still need to incorporate σ_y^2

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Adding Coefficients

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

```r
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
```

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) 606247.78  90734.64
## scale_sqft  330603.34  69914.18
##
## Error terms:
##  Groups   Name        Std.Dev.  Corr
##  zipcode  (Intercept) 271288.68
##           scale_sqft  208149.21 0.99
##  Residual             196377.64
## ---
## 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    330603.3
```

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

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

```
## $data
## [1] 196377.6
##
## $zipcode
## (Intercept)  scale_sqft
##    271288.7    208149.2
```

Final Connections

Hierarchical GLMs

Similar to the `lmer` syntax, `glmer` can be be used for multilevel generalized-linear models.

We will continue with the Seattle housing dataset and look to model whether a house has more than 2 bathrooms.

```
seattle <- seattle %>% mutate(more2 = bathrooms > 2, lessequal2 = bathrooms <= 2)
```

First look at the basic GLM with just an intercept.

```
glm(cbind(more2,lessequal2) ~ 1, data = seattle, family = binomial)
```

```
##
## Call:  glm(formula = cbind(more2, lessequal2) ~ 1, family = binomial,
##     data = seattle)
##
## Coefficients:
## (Intercept)
##     -0.1963
##
## Degrees of Freedom: 868 Total (i.e. Null);  868 Residual
## Null Deviance:       1196
## Residual Deviance: 1196  AIC: 1198
```

```
seattle %>% summarise(mean(more2))
```

```
## # A tibble: 1 x 1
##   `mean(more2)`
##           <dbl>
## 1         0.451
```

First look at the basic GLM with just an intercept.

```
glmer1 <- glmer(cbind(more2,lessequal2) ~ 1 + (1 | zipcode), data = seattle, family = binomial)
display(glmer1)
```

```
## glmer(formula = cbind(more2, lessequal2) ~ 1 + (1 | zipcode),
##     data = seattle, family = binomial)
## coef.est  coef.se
##    -0.12     0.21
##
## Error terms:
##  Groups   Name        Std.Dev.
##  zipcode  (Intercept) 0.60
##  Residual             1.00
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 1169.3, DIC = 1111
## deviance = 1138.2
```

```
fixef(glmer1)
```

```
## (Intercept)
##   -0.116311
```

```
ranef(glmer1)
```

```
## $zipcode
##       (Intercept)
## 98010 -0.14853329
## 98014 -0.13112403
## 98024  0.03705425
## 98032 -0.57833924
## 98039  1.30568826
## 98070 -0.49893697
## 98102  0.43437723
## 98109  0.05552176
## 98148 -0.47351847
##
## with conditional variances for "zipcode"
```

```
seattle %>% group_by(zipcode) %>% summarise(mean(more2))
```

```
## # A tibble: 9 x 2
##   zipcode `mean(more2)`
##   <fct>           <dbl>
## 1 98010           0.43
## 2 98014           0.435
## 3 98024           0.481
## 4 98032           0.32
## 5 98039           0.84
## 6 98070           0.339
## 7 98102           0.590
## 8 98109           0.486
## 9 98148           0.333
```

Covariates can also be added that vary across the groups

```
glmer2 <- glmer(cbind(more2,lessequal2) ~ 1 + bedrooms + (1 + bedrooms | zipcode),
                data = seattle, family = binomial)
display(glmer2)
```

```
## glmer(formula = cbind(more2, lessequal2) ~ 1 + bedrooms + (1 +
##     bedrooms | zipcode), data = seattle, family = binomial)
##             coef.est coef.se
## (Intercept) -4.00     0.61
## bedrooms     1.18     0.17
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  zipcode  (Intercept) 1.31
##           bedrooms    0.33     -0.95
##  Residual             1.00
## ---
## number of obs: 869, groups: zipcode, 9
## AIC = 1038, DIC = 966.9
## deviance = 997.4
```

```
sigma.hat(glmer2)$sigma$zipcode
```

```
## (Intercept)    bedrooms
##   1.3144794   0.3289755
```

```
fixef(glmer2)
```

```
## (Intercept)    bedrooms
##   -3.999126    1.184553
```

```
ranef(glmer2)
```

```
## $zipcode
##       (Intercept)    bedrooms
## 98010 -1.34602768  0.33724026
## 98014 -0.08880417  0.05982082
## 98024  0.48923371 -0.13855975
## 98032 -0.83039240  0.03273349
## 98039  0.29087670  0.09129365
## 98070  0.26759813 -0.09874033
## 98102  2.25163739 -0.52606428
## 98109  0.52726754 -0.11506187
## 98148 -1.15885121  0.24072636
##
## with conditional variances for "zipcode"
```

Stan & JAGS

There are a few other approaches for fitting these type of models. Stan and JAGS are two common (Bayesian) approaches for fitting hierarchical models. Both have additional flexibility for specifying sampling models directly.

Below is the syntax for a Stan model for hierarchical logistic regression.

```
data {
  int<lower=1> D;
  int<lower=0> N;
  int<lower=1> L;
  int<lower=0,upper=1> y[N];
  int<lower=1,upper=L> ll[N];
  row_vector[D] x[N];
}
parameters {
  real mu[D];
  real<lower=0> sigma[D];
  vector[D] beta[L];
}
model {
  for (d in 1:D) {
    mu[d] ~ normal(0, 100);
    for (l in 1:L)
      beta[l,d] ~ normal(mu[d], sigma[d]);
  }
  for (n in 1:N)
    y[n] ~ bernoulli(inv_logit(x[n] * beta[ll[n]]));
}
```