Homework 8

Austin Palmer

2023-04-19

15.1

A)

Complete Pooling: This is an approach to modeling grouped data where all observations in a group/dataset are combined into one average value. The data is treated as coming from the same entity, and their individual differences are ignored.

B)

No Pooling: Each observation in a group/dataset is treated as a distinct entity and accounted for in the model with a predictor variable.

C)

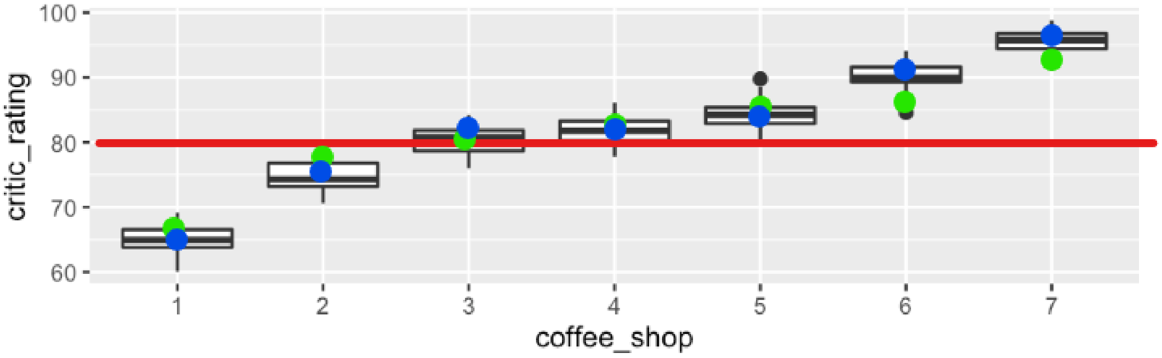
Partial Pooling: In between complete and no pooling, some level of data combination is used but groups are still allowed to have some degree of individuality and defining characteristics.

16.1

Red: Complete Pooled

Blue: No Pooled

Green: Hierarchical



16.6: Applied Exercises

```
big_word_club <- big_word_club %>%
  filter(treat == 1) %>%
  select(school_id, score_pct_change) %>%
  na.omit()
```

A)

```
head(big_word_club, 5)
```

```
##   school_id score_pct_change
## 1         2        -8.823529
## 2         2        14.285714
## 3         2        17.857143
## 4         2         6.666667
## 5         2        12.500000
```

```
length(unique(big_word_club$school_id))
```

```
## [1] 26
```

26 schools

B)

Range in num of student participates per school

```
school_stu_counts <-big_word_club %>%
  group_by(school_id) %>%
  count(school_id)

summary(school_stu_counts[2])
```

```
##        n
##  Min.   :12.00
##  1st Qu.:12.00
##  Median :12.50
##  Mean   :12.92
##  3rd Qu.:13.00
##  Max.   :17.00
```

```
sapply(school_stu_counts[2], sd, na.rm=TRUE)
```

```
##        n
## 1.293772
```

The number of students per participating school ranges between 12 and 17 with a standard deviation of 1.3 students. The mean and median are less than half a deviation apart suggesting the number of students per school could be appropriately described with a normal distribution.

C)

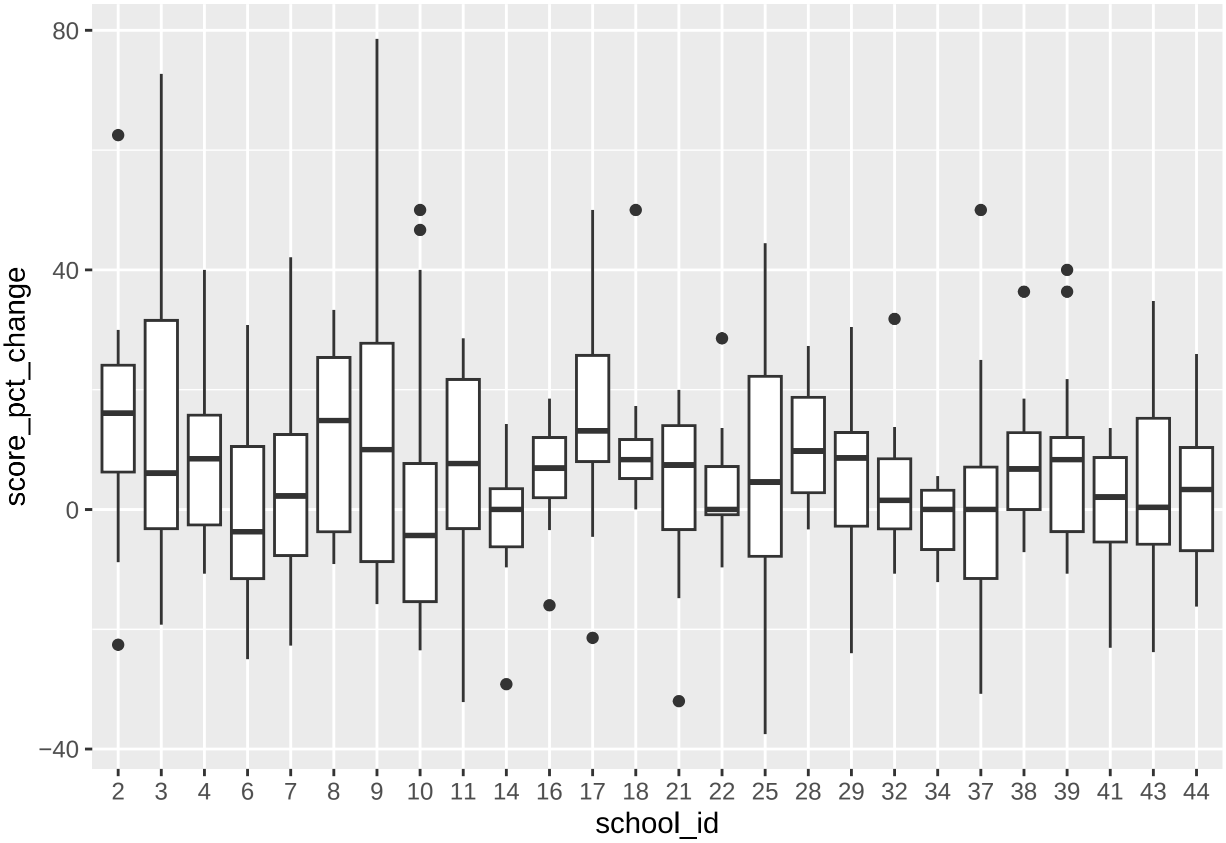
```
avg_school_improv <- big_word_club %>%
  group_by(school_id) %>%
  summarize(count = n(), score_pct_change = mean(score_pct_change))

avg_school_improv %>%
  slice(1:2, 25:26)
```

```
## # A tibble: 4 x 3
##   school_id count score_pct_change
##   <fct>     <int>            <dbl>
## 1 2            12            15.8
## 2 3            13            15.0
## 3 43           16             3.47
## 4 44           13             2.46
```

On average students at school id 2 displayed the largest score improvement (15.77%) and school id 44 the least at 2.45%

D)

```
ggplot(big_word_club, aes(x=school_id, y=score_pct_change)) +
  geom_boxplot()
```



The boxplot illustrates the degree to which some schools tend to improve more than others, as well as the variability in score improvements between students at the same school. We can see the groups are somewhat similar in mean, with nearly all being between 5 and 15, but the variability school to school swings wildly. School schools like $i = 25$ exhibit high consistency in score improvements and others like $j = 9$ range well more than double the 25th school.

16.7

A)

If we were to use a complete pooled approach we would have to define a global average score change parameters, resulting in the lose of individual characteristics of each school. Taking this route would leave us unable to compare and contrast within-school variability. Additionally, forcing this data to be fit by complete pool model would violate the model's independent observations assumption since we could expect students at the same school to exhibit similar features. On the other end of the extreme, using a no pooling model would leave us unable to generalize and predict for any school unconstrained in the sample. Additionally, the no pooling approach assumes that groups don't contain relevant information about one another, which could expect to be false since school material is very similar across schools, and would leave us unable to analyze the between-school variability. Modeling this data with a a hierarchical model allows us to examine both between- and within-group variability, and would take weight both the global mean and group mean into it's predictions.

B)

μ represents the average global score change in percent across all schools while μ_j represents a specific school's average score change in percent across their respective sample students.

C)

σ_y represents the within-group variability, i.e., the standard deviation between student scores within each school. σ_μ on the other hand, represents the between-group variability, i.e., the standard deviation in mean student score changes between schools.

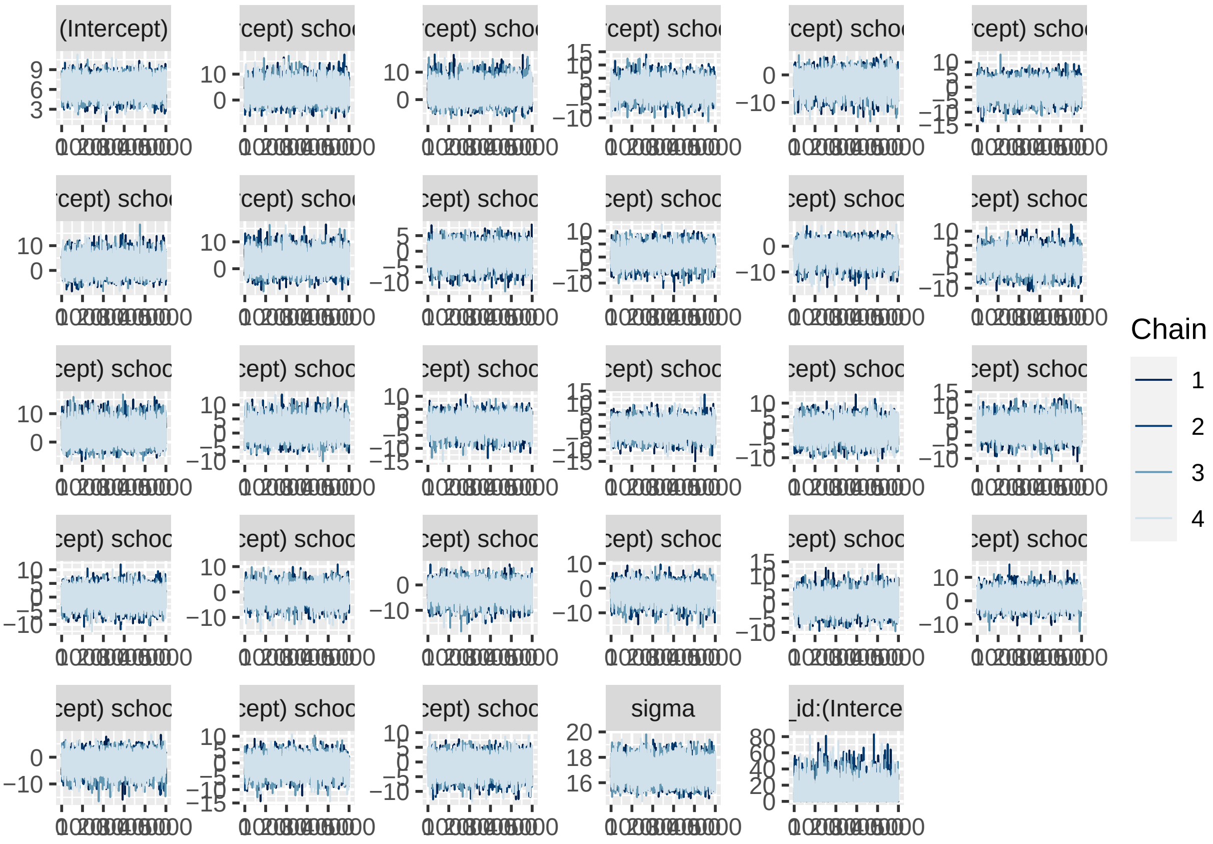
16.8

A)

```r
b0<-normal(8, 2.5, TRUE)
sigma<-exponential(1,TRUE)

scores_hierarchical_model <- stan_glmer(
score_pct_change ~ (1 | school_id),
data=big_word_club, family = gaussian,
prior_intercept = b0,
prior_aux = sigma,
prior_covariance = decov(reg=1, conc=1, shape=1, scale=1),
chains=4, iter=5000*2, seed=84735
)

prior_summary(scores_hierarchical_model)
```

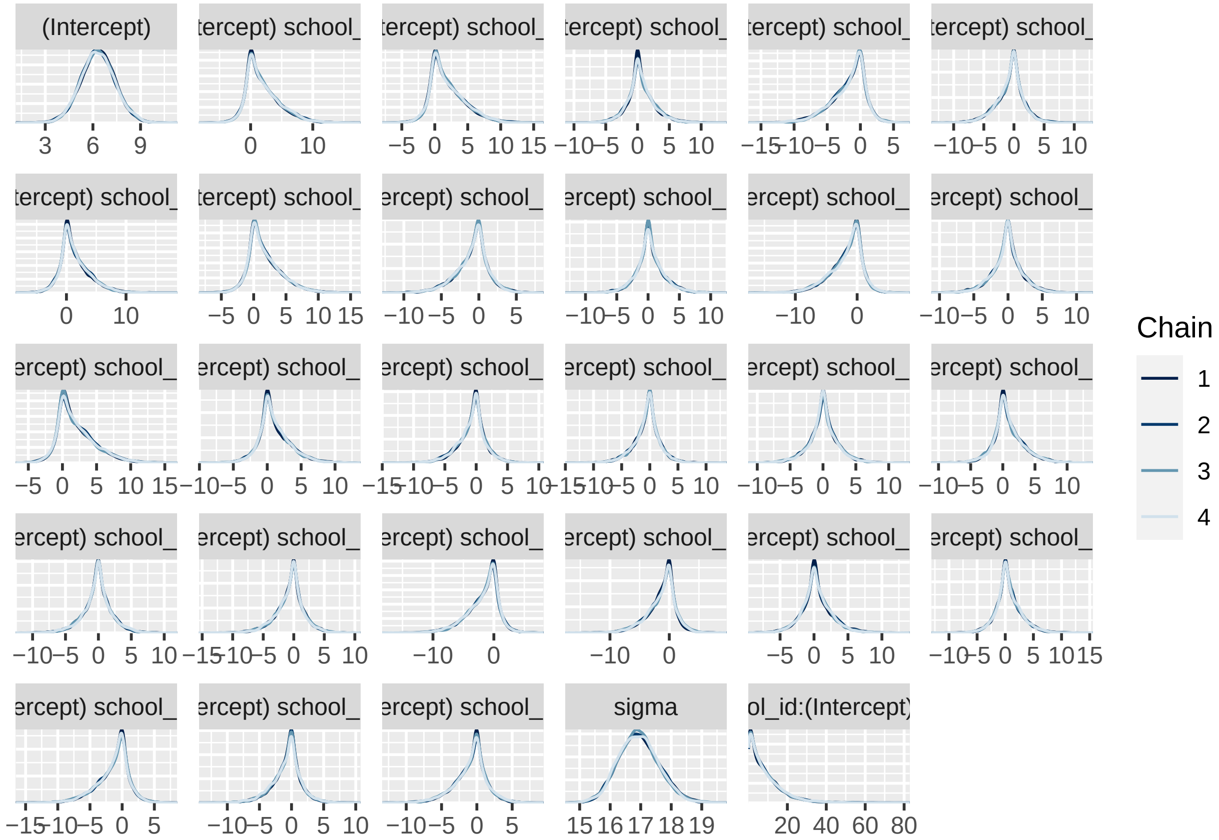
B)

```
mcmc_trace(scores_hierarchical_model)
```



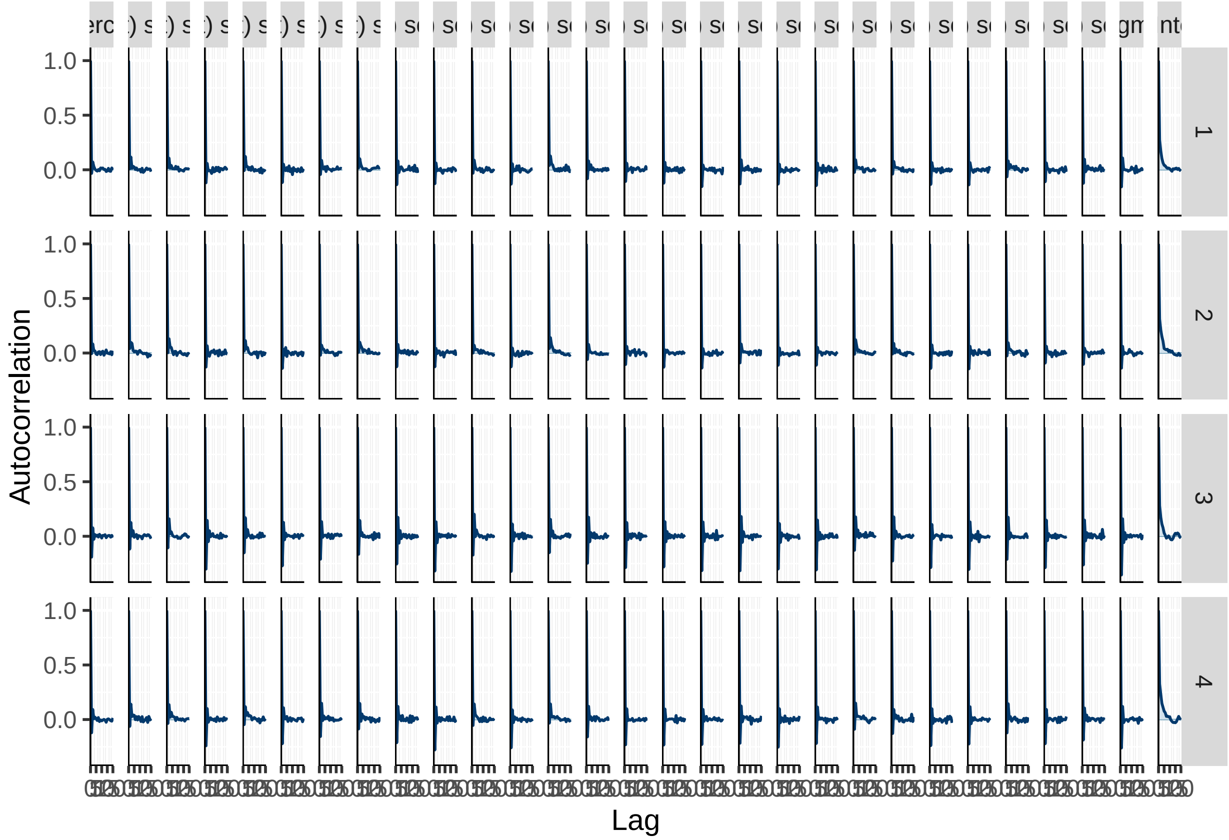
The trace plots appear to be stable and mixing well

```
mcmc_dens_overlay(scores_hierarchical_model)
```



All density overlays very similar, adding additional support that our model is fitting and mixing well.

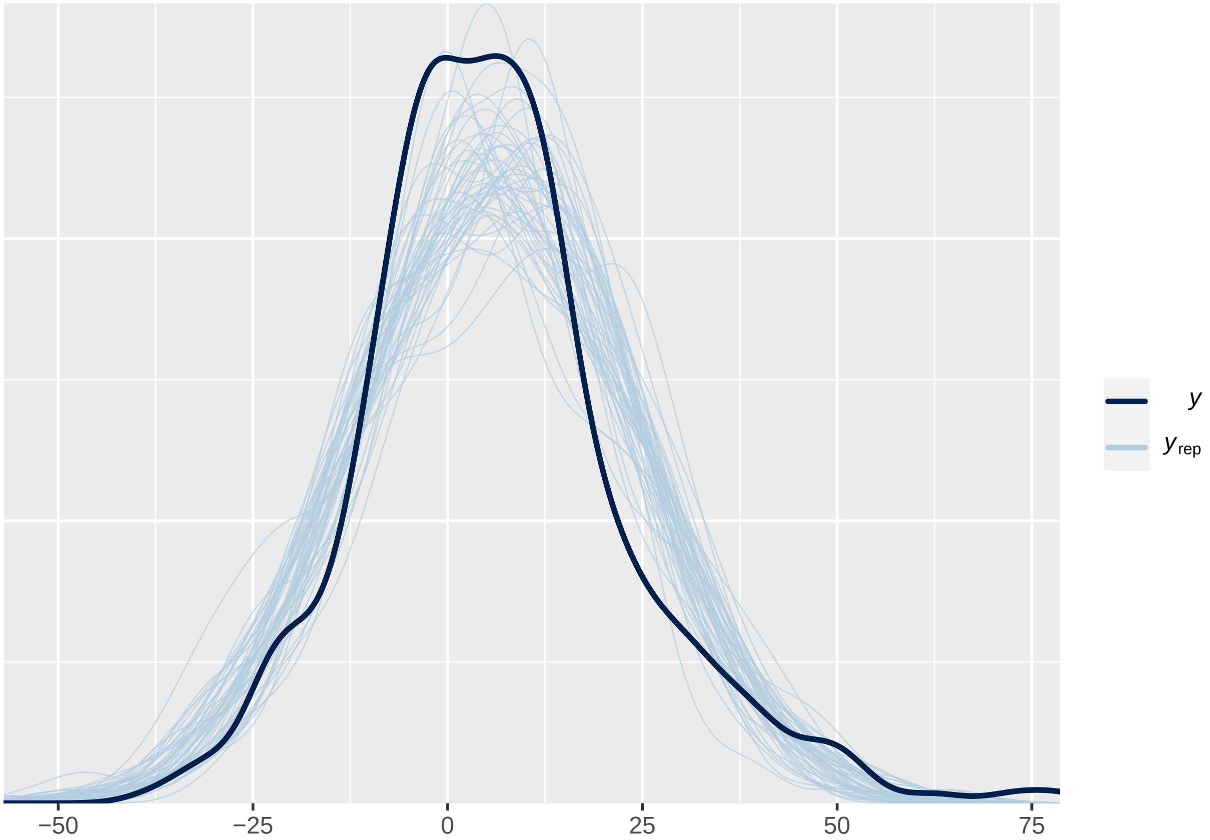
```r
mcmc_acf(scores_hierarchical_model)
```



All autocorrelation plots fall to 0 by the 5th step, supporting stabilization and a good mix.

C)

```
pp_check(scores_hierarchical_model)
```



Visually checking the posterior predictive check confirms that our hierarchical model isn't too wrong when it comes to capturing variability in score changes.

16.9

A)

```r
tidy(scores_hierarchical_model, effects = "fixed",
     conf.int = TRUE, conf.level = 0.80)
```

```
## # A tibble: 1 x 5
##   term        estimate std.error conf.low conf.high
##   <chr>          <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)     6.32      1.06     4.95      7.69
```

Per the results, there's an 80% chance that the average school percentage score change is between 4.95 and 7.69.

B)

There is ample evidence to say that on average, student vocabulary levels improve throughout the vocabulary program. The 80% credible posterior median for global μ is significantly positive and well above 0.

C)

```
tidy(scores_hierarchical_model, effects="ran_pars")
```

```
## # A tibble: 2 x 3
##   term                       group     estimate
##   <chr>                      <chr>        <dbl>
## 1 sd_(Intercept).school_id school_id     2.88
## 2 sd_Observation.Residual  Residual     16.9
```

```
2.88^2/(2.88^2+16.94^2)
```

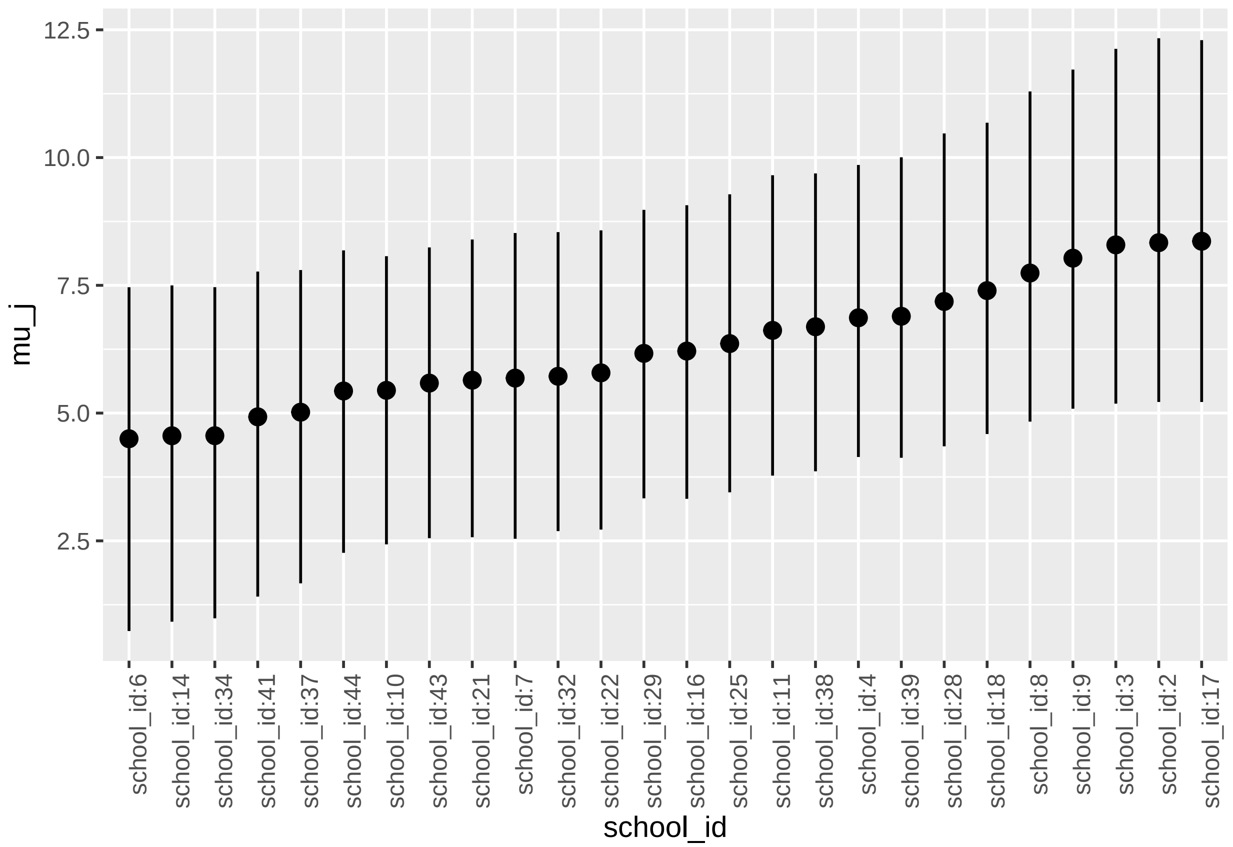
```
## [1] 0.02809204
```

The posterior median of σ_y suggests that within any given school, percent score changes tend to vary by 2.88 from student to student. The between standard deviation tends to be much higher at around 16.94, interpreted as the mean percent score change tends to vary by 16.94 from school to school. From these values we find that the percent changes in score by students of the same school tend to have have very little correlation near 0.03. This can be interpreted to mean that 3% of the variation in percent score change is explained by differences between schools, whereas 97% is explained by differences among the students within each school.

16.10

A)

```
# Get MCMC chains for each mu_j
score_chains <- scores_hierarchical_model %>%
  spread_draws(`(Intercept)`, b[,school_id]) %>%
  mutate(mu_j = `(Intercept)` + b)

# Get posterior summaries for mu_j
score_summary_scaled <- score_chains %>%
  select(-`(Intercept)`, -b) %>%
  mean_qi(.width = 0.80) %>%
  mutate(school_id = fct_reorder(school_id, mu_j))

# Check out the results
ggplot(score_summary_scaled,
       aes(x = school_id, y = mu_j, ymin = .lower, ymax = .upper)) +
  geom_pointrange() +
  xaxis_text(angle = 90, hjust = 1)
```



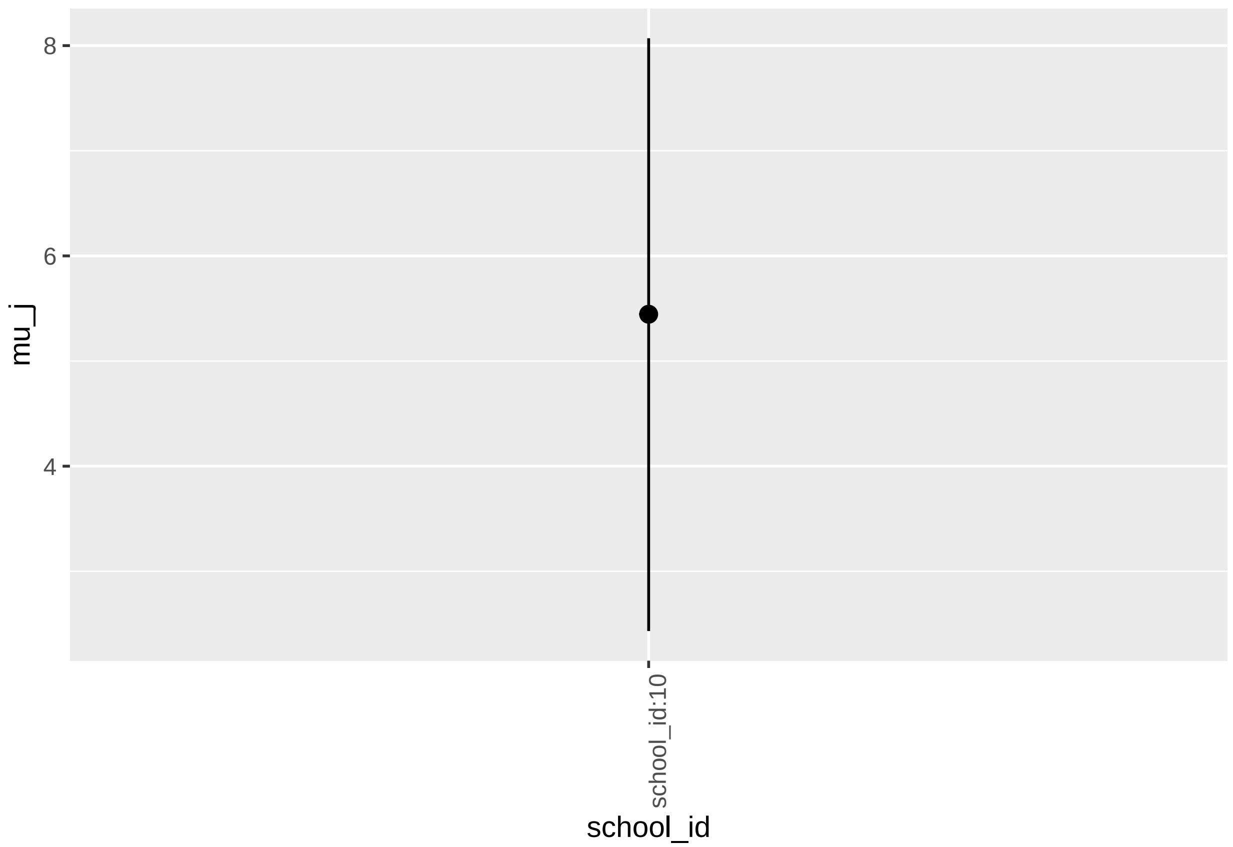
Plotting 80% posterior credible intervals for all schools in our sample we can see that μ_j varies much more in location than by scale. Most schools have a similar range credible range of mean scores by students but their mean, μ_j, varies significantly school to school.

B)

```
j10 <- score_summary_scaled[which(score_summary_scaled$school_id == "school_id:10"),]
j10
```

```
## # A tibble: 1 x 7
##   school_id     mu_j .lower .upper .width .point .interval
##   <fct>        <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
## 1 school_id:10  5.45   2.43   8.07    0.8 mean   qi
```

```
ggplot(j10,
       aes(x = school_id, y = mu_j, ymin = .lower, ymax = .upper)) +
  geom_pointrange() +
  xaxis_text(angle = 90, hjust = 1)
```



For $\mu_1 0$, we are 80% sure `school_id=25` has a mean average student percent score change between 3.45 and 9.28.

C)

There is not ample evidence that on average vocab scores at school 10 improved by more than 5%. Although 5 in contained in the credible interval and is 3.5 is the lower bound so it's very plausible that vocab scores at School 10 will improve less than 5% during the program.

16.11

A)

School 6

```
sch6 <- score_summary_scaled[which(score_summary_scaled$school_id == "school_id:6"),]
sch17 <- score_summary_scaled[which(score_summary_scaled$school_id == "school_id:17"),]

scores_df <- as.data.frame(scores_hierarchical_model)
sch6_chains <- scores_df %>%
  rename(b = `b[(Intercept) school_id:6]`) %>%
  select(`(Intercept)`, b, sigma) %>%
  mutate(mu_6 = `(Intercept)` + b,
         y_6 = rnorm(20000, mean = mu_6, sd = sigma))
```

```
# Check it out
head(sch6_chains, 6)
```

```
##   (Intercept)           b    sigma      mu_6         y_6
## 1    6.655358 -5.92918034 17.21016 0.7261777  35.6382954
## 2    6.048751  0.72741243 16.52209 6.7761631  11.7050477
## 3    6.455060 -0.34551708 16.52370 6.1095426 -15.3886863
## 4    6.949515 -0.20363586 16.99411 6.7458791 -12.1740107
## 5    6.610420  0.22681123 16.92273 6.8372308  -0.7471881
## 6    5.971805  0.09562842 17.36304 6.0674331   7.7185352
```

Bayes Prep

```
bayes_chains <- scores_df %>%
  mutate(sigma_mu = sqrt(`Sigma[school_id:(Intercept),(Intercept)]`),
         mu_bayes = rnorm(20000, `(Intercept)`, sigma_mu),
         y_bayes = rnorm(20000, mu_bayes, sigma))
```

```
# Check it out
head(bayes_chains, 6)
```

```
##   (Intercept) b[(Intercept) school_id:2] b[(Intercept) school_id:3]
## 1    6.655358                 8.76047731                2.697417729
## 2    6.048751                -0.01023321                1.046096979
## 3    6.455060                 0.81131605                0.507479570
## 4    6.949515                -0.78769405                0.213151299
## 5    6.610420                -1.40116558               -1.210542439
## 6    5.971805                -0.04036963                0.002496569
##   b[(Intercept) school_id:4] b[(Intercept) school_id:6]
## 1                 -2.2636348                -5.92918034
## 2                  1.6518946                 0.72741243
## 3                  0.1296125                -0.34551708
## 4                  0.7786082                -0.20363586
## 5                  0.9864283                 0.22681123
## 6                 -0.1835984                 0.09562842
##   b[(Intercept) school_id:7] b[(Intercept) school_id:8]
## 1                 -3.4442224                 0.94785739
## 2                  0.8320097                 1.38532716
## 3                 -0.4412931                 0.51509961
## 4                 -0.1681463                -0.06114488
## 5                  0.3629896                -0.42454889
## 6                  0.0298865                -0.08094325
##   b[(Intercept) school_id:9] b[(Intercept) school_id:10]
## 1                 9.00635936                  -2.7027285
## 2                -1.62503954                  -0.2091193
## 3                -0.61232332                   0.7304434
## 4                 1.60594732                  -1.8433018
## 5                 2.90078746                  -2.8828183
## 6                 0.08967748                  -0.1534465
##   b[(Intercept) school_id:11] b[(Intercept) school_id:14]
## 1                 2.276672557                 -1.70593774
## 2                -0.735964461                 -1.86612634
## 3                 0.000869798                 -0.06795942
## 4                 0.360011220                 -1.54881596
## 5                -0.743619342                 -1.89733357
## 6                -0.034768652                  0.02235619
##   b[(Intercept) school_id:16] b[(Intercept) school_id:17]
## 1                  -4.8672190                  4.61198724
## 2                   1.7180913                  0.99745608
## 3                   0.2532084                 -0.25645831
## 4                  -0.8257425                  1.68636022
## 5                  -1.1244320                  1.52879674
## 6                  -0.1180102                 -0.03931344
##   b[(Intercept) school_id:18] b[(Intercept) school_id:21]
## 1                  4.72315221                 -6.52242524
## 2                 -1.43852443                  2.84359709
## 3                 -0.33672771                 -0.05643249
## 4                  0.94222673                 -0.26749130
## 5                  1.68739666                 -0.59384664
## 6                 -0.09188585                  0.19695565
##   b[(Intercept) school_id:22] b[(Intercept) school_id:25]
## 1                  -3.2366237                   2.4625071
## 2                   0.7563910                  -1.0079319
## 3                  -0.1978403                  -0.2967312
## 4                  -0.6645447                   0.4933069
## 5                  -1.3313314                  -0.6006120
## 6                   0.1104300                  -0.1553381
##   b[(Intercept) school_id:28] b[(Intercept) school_id:29]
## 1                 -0.87994164                   0.5482891
## 2                  0.92179709                  -0.3421078
## 3                  0.25651274                   0.4448916
## 4                  0.52316315                  -1.1290665
## 5                  1.66733539                  -1.6265735
## 6                 -0.06947768                  -0.2083939
##   b[(Intercept) school_id:32] b[(Intercept) school_id:34]
## 1                  -1.9010313                 -2.12491505
## 2                  -0.0585468                 -1.84590539
## 3                   0.4708896                 -0.32002492
## 4                  -1.7253709                 -0.79543073
## 5                  -2.2943281                 -1.41761920
## 6                   0.1005498                 -0.09922648
##   b[(Intercept) school_id:37] b[(Intercept) school_id:38]
## 1                 -1.82496285                   5.6537991
## 2                 -1.55315390                  -2.2568003
## 3                 -0.84825343                  -0.5070147
## 4                  0.07832322                   1.6217799
## 5                  1.48477428                   2.5311592
## 6                  0.22141553                  -0.1179946
##   b[(Intercept) school_id:39] b[(Intercept) school_id:41]
## 1                  -5.6014351                 -4.28978285
## 2                   2.7494406                  0.01319492
## 3                   0.7384752                 -0.34657543
## 4                  -1.0845565                  0.94909428
## 5                  -2.0174781                  1.20414072
## 6                   0.2225251                 -0.00781081
##   b[(Intercept) school_id:43] b[(Intercept) school_id:44]    sigma
## 1                  2.51966873                    1.1380535 17.21016
## 2                 -1.04191168                   -1.9212554 16.52209
## 3                  0.03585412                   -0.3108822 16.52370
## 4                 -0.27580764                    0.7735944 16.99411
## 5                 -1.43822595                    1.8321492 16.92273
## 6                 -0.08594148                   -0.1159177 17.36304
##   Sigma[school_id:(Intercept),(Intercept)]  sigma_mu  mu_bayes    y_bayes
## 1                              16.64548600 4.0798880 10.099259  16.166111
## 2                               2.77517943 1.6658870  6.604770  -3.144348
## 3                               0.22544107 0.4748063  5.798554  25.197065
## 4                               0.88114814 0.9386949  5.311295 -27.113906
## 5                               1.46154788 1.2089449  8.203626  12.594893
## 6                               0.01846443 0.1358839  6.043395   6.304829
```

B)

```
# Posterior summary of Y_new,j
sch6_chains %>%
  mean_qi(y_6, .width = 0.80)
```

```
## # A tibble: 1 x 6
##     y_6 .lower .upper .width .point .interval
##   <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
## 1  4.47  -17.3   26.5    0.8 mean   qi
```

The 80% posterior predictive interval of $Y_{new,j}$ for School 6 is centered at 4.62 and ranges from -17.13 to 26.49.

```
# Posterior summary of Y_new,j
bayes_chains %>%
  mean_qi(y_bayes, .width = 0.80)
```

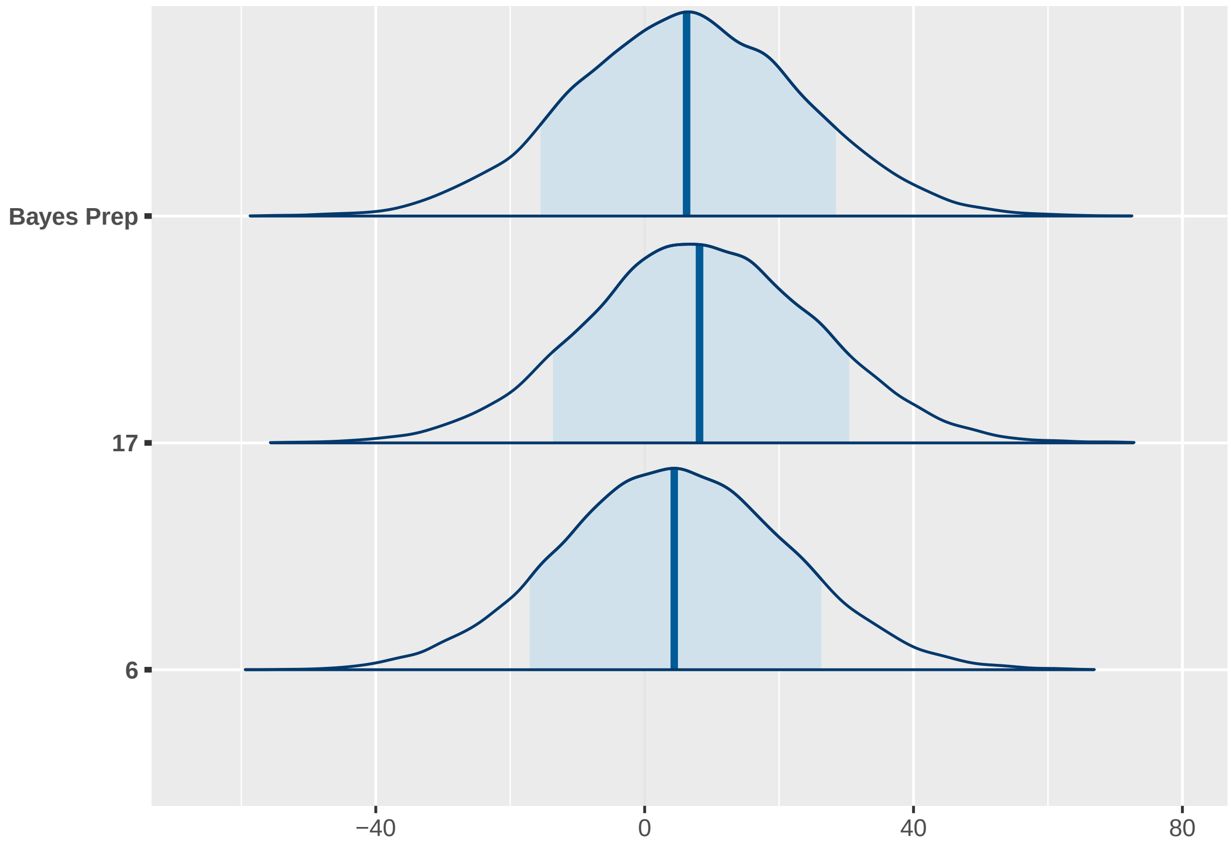
```
## # A tibble: 1 x 6
##   y_bayes .lower .upper .width .point .interval
##     <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
## 1    6.37  -15.6   28.3    0.8 mean   qi
```

The 80% posterior predictive interval of $Y_{new,j}$ for Bayes Prep is centered at 6.26 and ranges from -15.58 to 28.09. Bayes Prep is predicted two percent higher than school 6, which follows with the data since School 6 is the worth performing school and Bayes Prep is unknown to our model so we could expect it's predicted score change to be closer to the mean than the lowest school. The deviation ranges are very similar in magnitude as well.

C)

```
set.seed(84735)


sch6_predictions <- posterior_predict(
  scores_hierarchical_model,
  newdata = data.frame(school_id = c(6,17, "Bayes Prep")))

# Posterior predictive model plots
mcmc_areas(sch6_predictions, prob = 0.8) +
  ggplot2::scale_y_discrete(labels = c("6", "17", "Bayes Prep"))
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

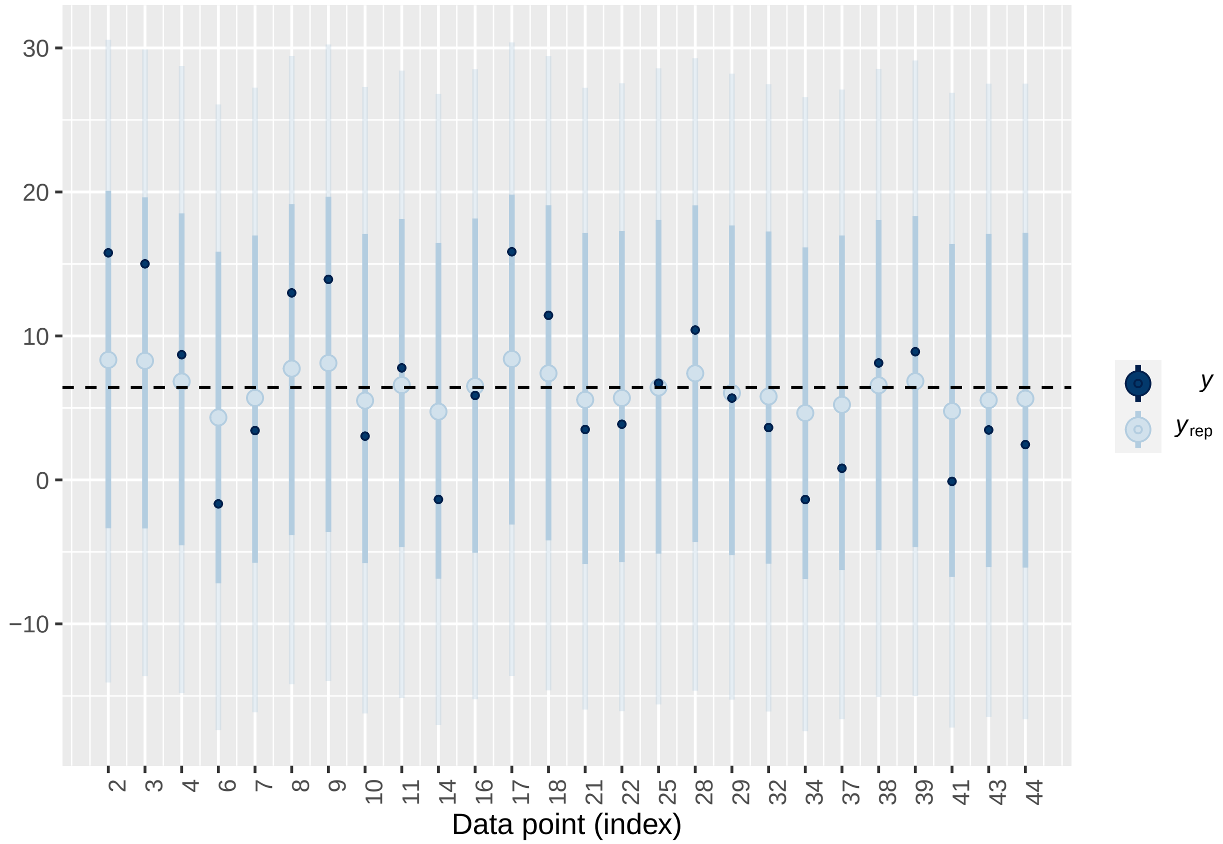


We anticipate that School 17 will have the highest percent change in score, School 6 the lowest, and Bayes Prep will fall in the middle. This follows logically as School 6 and School 17 were the lowest and highest averages respectively, and Bayes Prep was predicted without data and such should be closer to the mean than the two extreme cases.

D)

```
bwc_means <- big_word_club %>%
  group_by(school_id) %>%
  summarize(count = n(), score_pct_change = mean(score_pct_change))

gbl_mean <- mean(bwc_means$score_pct_change)
```

```
set.seed(84735)
predictions_hierarchical <- posterior_predict(scores_hierarchical_model,
                                             newdata = bwc_means)

# Posterior predictive plots
ppc_intervals(bwc_means$score_pct_change, yrep = predictions_hierarchical,
              prob_outer = 0.80) +
  ggplot2::scale_x_continuous(labels = bwc_means$school_id,
                              breaks = 1:nrow(bwc_means)) +
  xaxis_text(angle = 90, hjust = 1) +
  geom_hline(yintercept = gbl_mean, linetype = "dashed")
```



From the 80% posterior plot for all schools in our sample, the no pooled model predictions (dark blue) are consistently further away from the global mean (dashed line) than the hierarchical predictions (light blue). We can say that the model is exhibiting significant shrinkage as all hierarchical predictions are significantly weighted toward the global mean and away from the group specific mean. This follows with what we found earlier in the composition of variance where between-school variance accounted for only 3% of the total while within-school variance contributed 97%. The heavy weight toward variability within groups implies that the model is heavily relying on global trends because there is little distinction in the patterns from one school to the next.