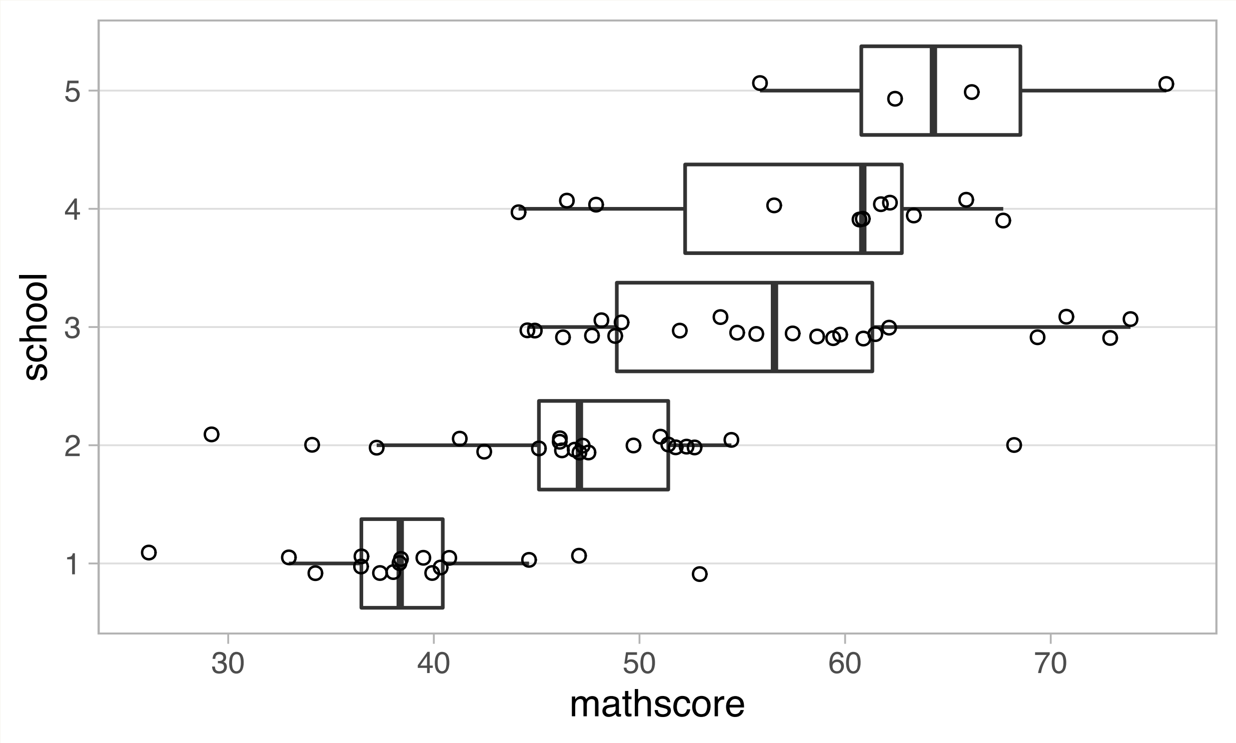
Hierarchical models

Stat 340: Bayesian Statistics

Example: ELS math scores

- 2002 Educational Longitudinal Study (ELS)
- Survey from schools across the United States
- Data are collected by sampling schools and then sampling students within each selected school
- We'll focus on 10th grade math scores from a sample of 10 schools
- Math tests contained items in arithmetic, algebra, geometry, data/probability, and advanced topics were divided into process categories of skill/knowledge, understanding/ comprehension, and problem solving

ELS math scores



Possible questions:

- What's the typical math score?
- To what extent do scores vary from school to school?
- For any single school, how much might scores vary from student to student?

Possible analysis strategies

Complete pooling (combined estimates)
Ignore schools and lump all students together

No pooling (separate groups)
Separately analyze each school and assume that one school's data doesn't contain valuable information about another school

Partial pooling (compromise estimates)
Acknowledge the grouping structure, so that even though schools differ in performance, they might share valuable information about each other and about the broader population of schools

What have we seen so far?

- Completely pooled model does not acknowledge differences between schools
- No pooled model acknowledges that some schools tend to score higher than others
- No pooled model ignores data on one school when learning about the typical score of another
- No pooled model cannot be generalized to schools outside our sample

Hierarchical model

Let's compromise between the the complete pooled and no pooled models

How? By using a *two-stage prior* specification

Hierarchical model specification for JAGS

```
modelString <-"model {

## sampling
for (i in 1:N){
   y[i] ~ dnorm(mu_j[school[i]], invsigma2)
}

## priors
for (j in 1:J){
   mu_j[j] ~ dnorm(mu, invtau2)
}

invsigma2 ~ dgamma(a_s, b_s)
sigma <- sqrt(pow(invsigma2, -1))

## hyperpriors
mu ~ dnorm(mu0, g0)
invtau2 ~ dgamma(a_t, b_t)
tau <- sqrt(pow(invtau2, -1))
}
"
```

Define the data and prior parameters

```r
y <- sub_school$mathscore
school <- sub_school$school
N <- length(y)
J <- length(unique(school))
the_data <- list(y = y, school = school,
                 N = N, J = J,
                 mu0 = 50, g0 = .04,  # prior parameters
                 a_t = 1, b_t = .01,  # hyperparameters
                 a_s = 1, b_s = .01)  # hyperparameters
```

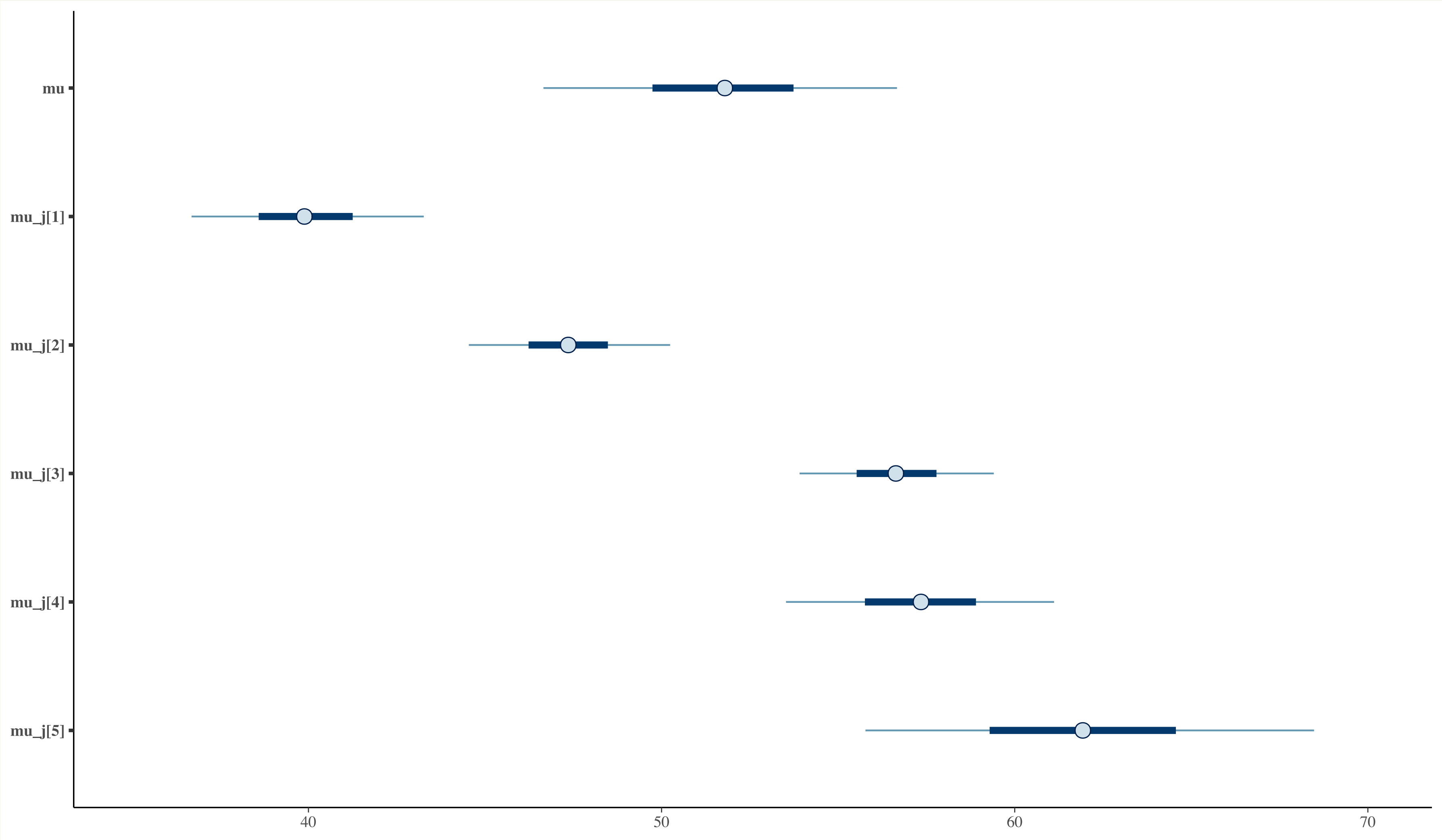
Run MCMC

```r
posterior <- run.jags(
  modelString,
  n.chains = 1,
  data = the_data,
  monitor = c("mu", "tau", "mu_j", "sigma"),
  adapt = 1000,
  burnin = 5000,
  sample = 5000,
  silent.jags = TRUE
)
```

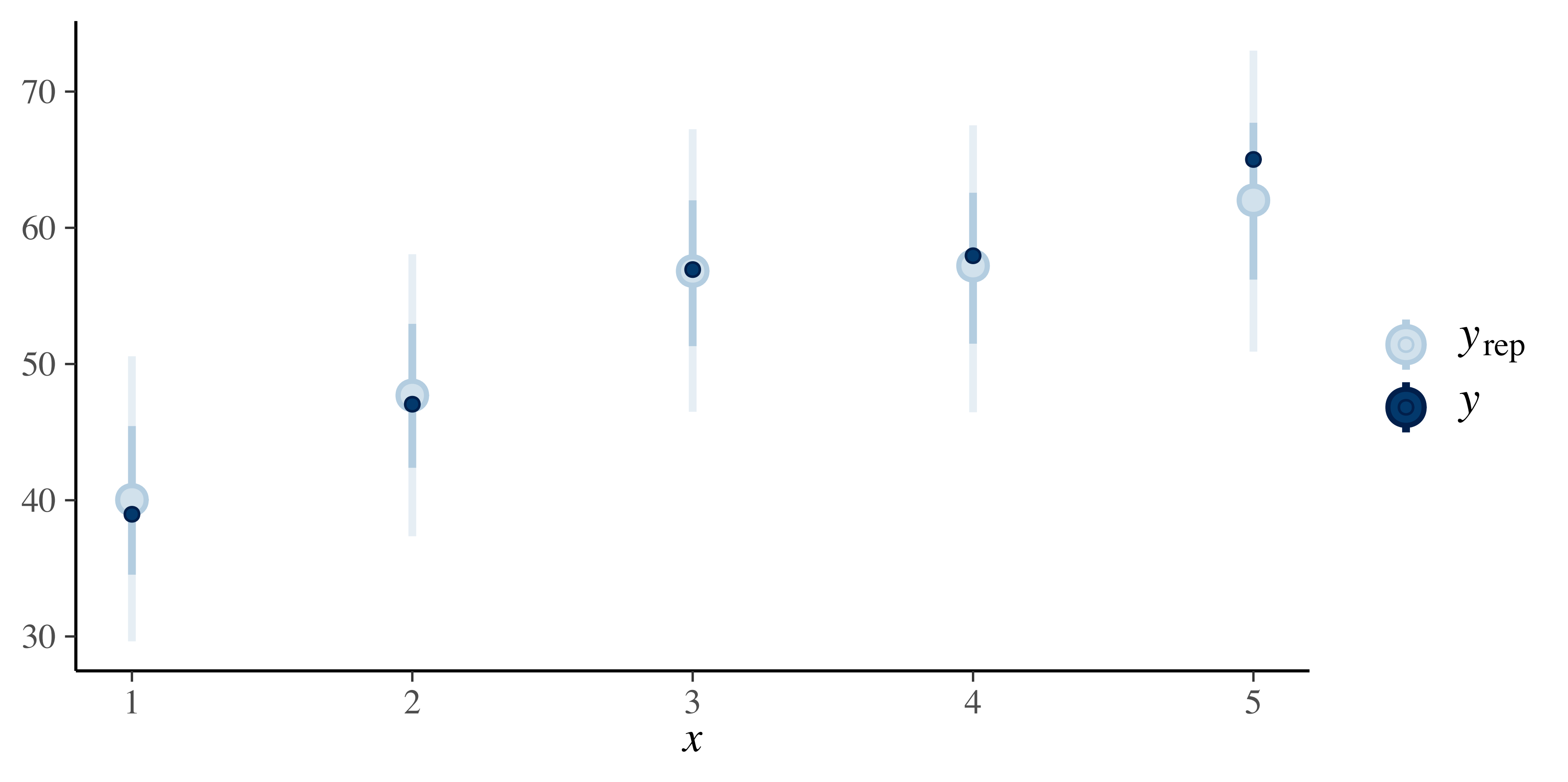
```r
print(posterior, digits = 3)
```

```
##
## JAGS model summary statistics from 5000 samples (adapt+burnin = 6000):
##
##          Lower95 Median Upper95 Mean     SD Mode  MCerr MC%ofSD SSeff     AC.10
## mu          45.7   51.8    57.9 51.7   3.06 51.9 0.0474     1.6  4156   -0.0257
## tau         3.62   7.61    14.1 8.19   3.03 6.87 0.0538     1.8  3178   -0.0105
## mu_j[1]     36.1   39.9    43.8 39.9      2 39.9 0.0319     1.6  3922   0.00146
## mu_j[2]     44.1   47.4    50.8 47.4   1.69 47.3 0.0241     1.4  4960   0.00413
## mu_j[3]     53.6   56.6      60 56.7   1.67 56.7 0.0236     1.4  4978    0.0174
## mu_j[4]     52.7   57.3    61.7 57.3   2.32 57.6 0.0344     1.5  4549    -0.015
## mu_j[5]     54.5   61.9    69.3   62   3.88 62.2 0.0651     1.7  3555   -0.0172
## sigma       6.68   7.92    9.28 7.97  0.683 7.83 0.0106     1.6  4121 -0.000238
##
##          psrf
## mu         --
## tau        --
## mu_j[1]    --
## mu_j[2]    --
## mu_j[3]    --
## mu_j[4]    --
## mu_j[5]    --
## sigma      --
##
## Total time taken: 0.9 seconds
```

```r
mcmc_intervals(posterior$mcmc, regex_pars = "mu")
```

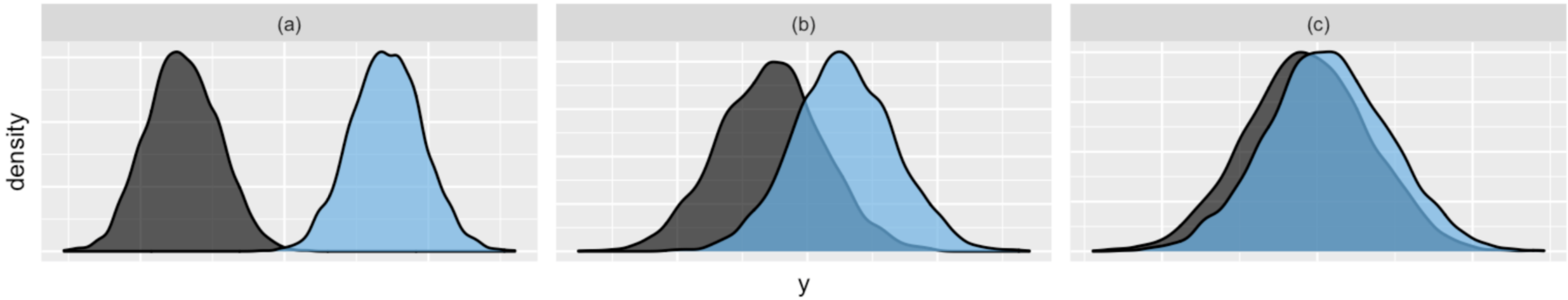


Hierarchical predictions vs. sample means



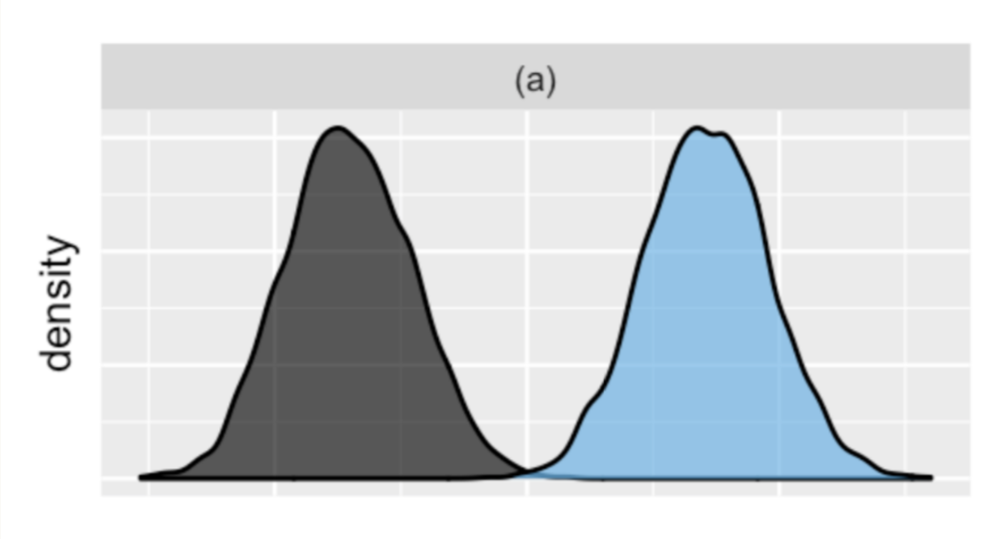
Comparing sources of variability

between > within | between ≈ within | between < within



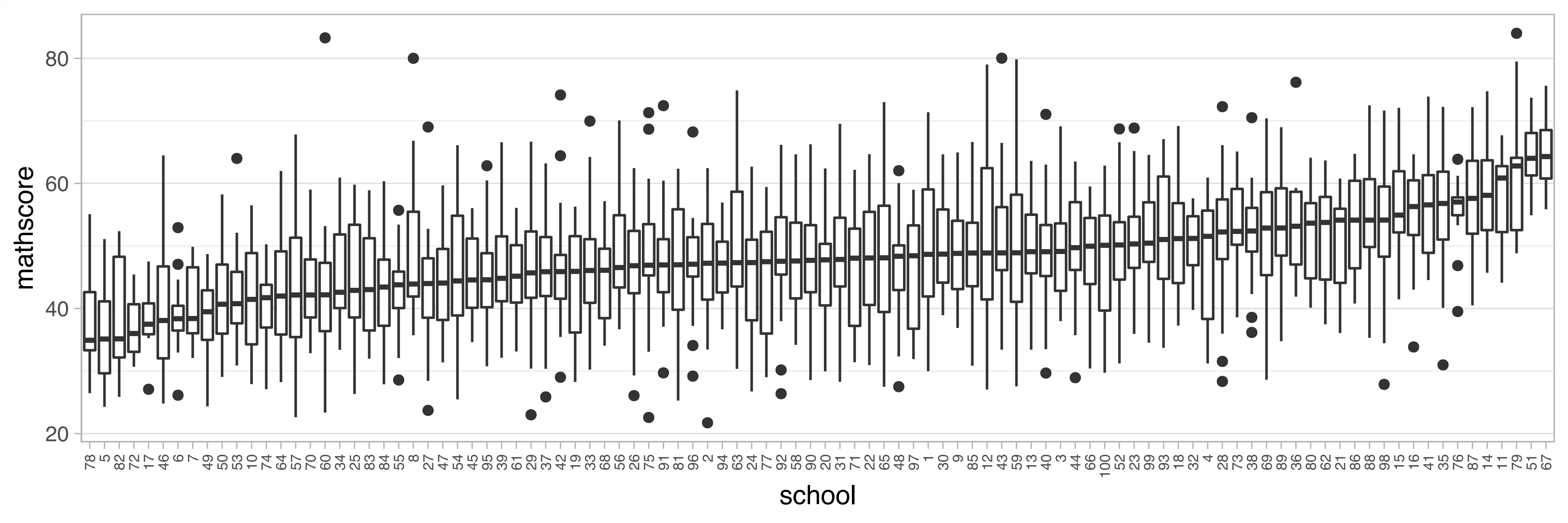
Within-group (intraclass) correlation

Suppose we're in the situation where between group variability is much larger than within group variability



- Two observations within the same group are more similar than two observations from different groups
- Observations within the same group are correlated (generally true, easier to see in this extreme case)

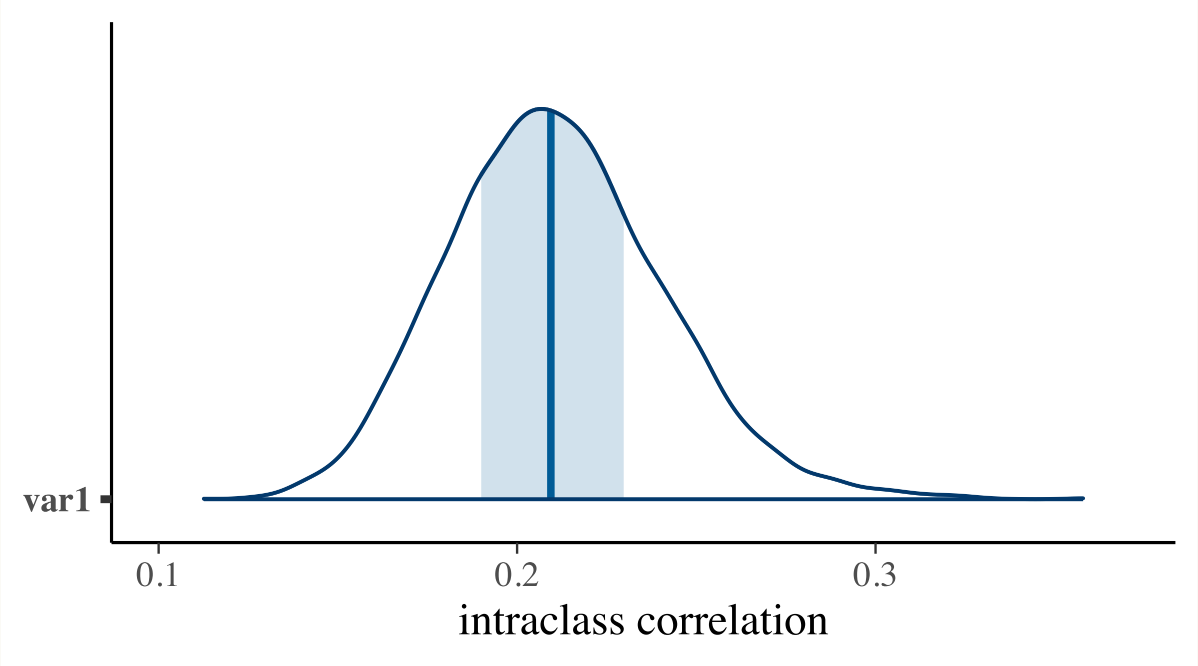
Full ELS data set



- 100 schools from urban settings in the full data set
- Sample sizes range from 4 to 32 students

ELS intraclass correlation

```
draws <- posterior_full$mcmc[[1]]
icc <- draws[,"tau"]^2 / (draws[,"tau"]^2 + draws[,"sigma"]^2)
```

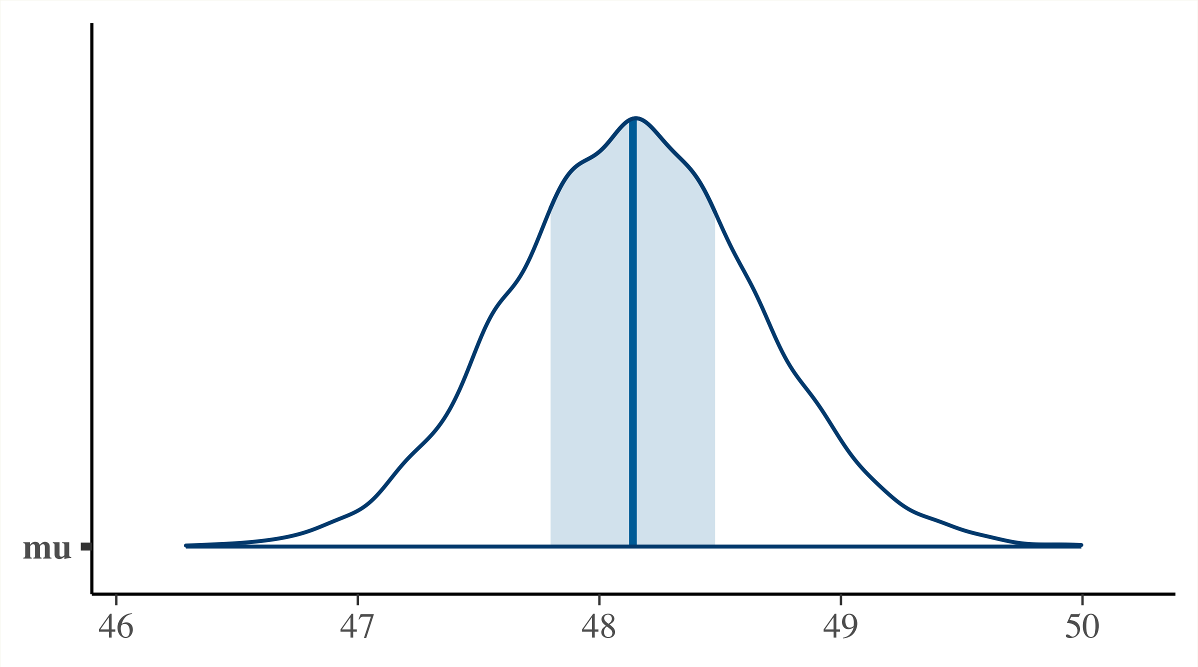


90% credible interval:

```
##      5%   95%
## 0.164 0.262
```

ELS global mean

Inference for the global parameters proceeds as always

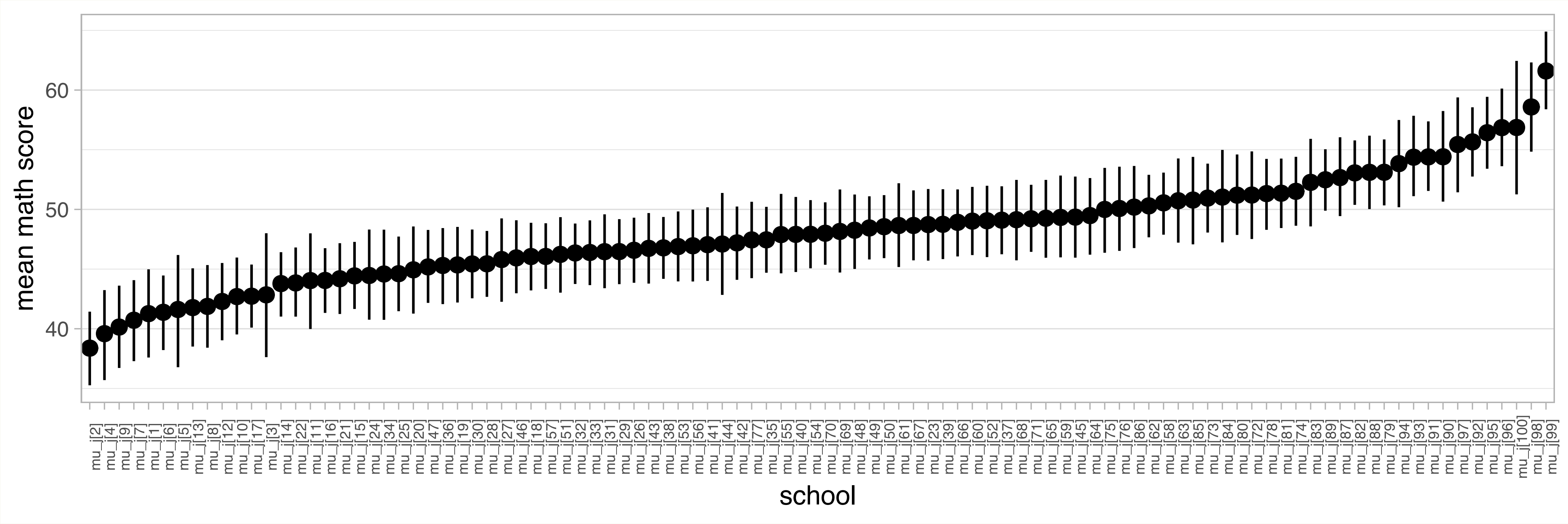


90% credible interval:

```
##     5%   95%
## 47.28 48.99
```

Inference for group-specific means

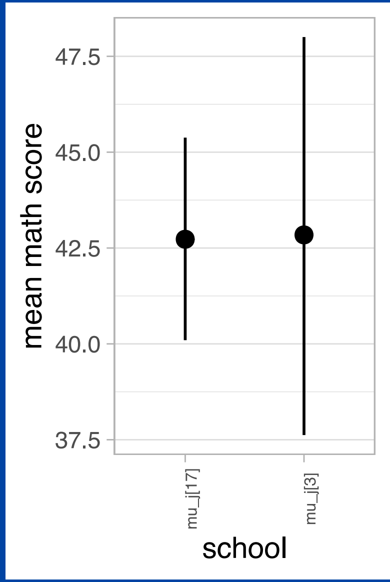
There are often *a lot* of parameters to manage for group-specific inference



Your turn

School 3 and 17 have roughly the same posterior mean, but substantially different credible interval widths

Discuss with a neighbor why you think this difference occurs.



02:00

Prediction for observed group

Suppose we want to make a prediction for school 13, then we need a posterior predictive distribution

```
# Work with a data frame
post_df <- as.data.frame(posterior_full$mcmc[[1]])

# Create the posterior predictive via simulation
pp_school13 <- post_df %>%
  select(mu_j = "mu_j[13]", sigma) %>%                # select relevant cols
  mutate(y_pred = rnorm(nrow(post_df), mu_j, sigma)) # simulate a new obs.

# Check it out
head(pp_school13, 3)
```

```
##          mu_j    sigma   y_pred
## 6001 41.04391 9.348647 17.76017
## 6002 43.43353 9.285637 52.01264
## 6003 41.57400 9.101547 38.62476
```

Prediction for unobserved group

Suppose we want to make a prediction for a school we didn't observe, let's call it school 101

```r
# Create the posterior predictive via simulation
pp_school101 <- post_df %>%
  select(mu, tau, sigma) %>%                          # select global params
  mutate(
    mu_j = rnorm(nrow(post_df), mu, tau),       # generate mu_j
    y_pred = rnorm(nrow(post_df), mu_j, sigma) # generate y
  )

# Check it out
head(pp_school101, 3)
```

```
##               mu      tau    sigma     mu_j   y_pred
## 6001 48.98863 4.967225 9.348647 40.99028 51.17038
## 6002 48.36430 4.651078 9.285637 46.81391 39.47738
## 6003 47.56924 5.185095 9.101547 41.29647 42.03361
```

How do the predictions compare?