STA 610L: Module 2.5

Random effects ANOVA (Bayesian estimation II)

Dr. Olanrewaju Michael Akande

STA 610L

Bayesian random effects ANOVA model

- Recall our hierarchical model can be written as

$$\begin{aligned} y_{ij}|\mu_j, \sigma^2 &\sim \mathcal{N}\left(\mu_j, \sigma^2\right); \quad i = 1, \ldots, n_j \\ \mu_j|\mu, \tau^2 &\sim \mathcal{N}\left(\mu, \tau^2\right); \quad j = 1, \ldots, J, \end{aligned}$$

with priors

$$\begin{aligned} \pi(\mu) &= \mathcal{N}\left(\mu_0, \gamma_0^2\right) \\ \pi(\tau^2) &= \mathcal{IG}\left(\frac{\eta_0}{2}, \frac{\eta_0 \tau_0^2}{2}\right) \\ \pi(\sigma^2) &= \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right). \end{aligned}$$

- We can write our own Gibbs sampler for this model (see end of this module and also, next homework).

- However, since we will rely primarily on the brms package for fitting many of our hierarchical models anyway, let's see if we can fit a version of this model to the radon data, using the brms package.

STA 610L

Radon study again

```r
#library(rstan)
#library(brms)
#library(tidybayes)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

#note: there are many ways of specifying priors in brms
#we will touch on some other options soon
#see the help page for "set_priors"
#for now, vague priors under our model specification will do
prior <- c(set_prior("normal(0,5)", class = "Intercept"),
           #set_prior("normal(0,10)", class = "b"),
           set_prior("inv_gamma(0.5,5)", class = "sigma"),
           set_prior("inv_gamma(0.5,5)", class = "sd"))

m1 <- brm(log_radon ~ (1 | countyname),
          data = Radon, family = gaussian(),
          prior = prior,iter = 3000, warmup = 2000,seed = 13)

summary(m1)
```

STA 610L

Radon study again

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: log_radon ~ (1 | countyname)
##    Data: Radon (Number of observations: 919)
##   Draws: 4 chains, each with iter = 5000; warmup = 2000; thin = 1;
##          total post-warmup draws = 12000
##
## Group-Level Effects:
## ~countyname (Number of levels: 85)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.40      0.05     0.31     0.51 1.00     5818     7459
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.32      0.06     1.21     1.43 1.00     7103     8310
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.80      0.02     0.76     0.84 1.00    22035     9252
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

STA 610L

Radon study again

We can compare the results to the frequentist estimates.

```
Model1 <- lmer(log_radon ~ (1 | countyname), data = Radon)
summary(Model1)
```

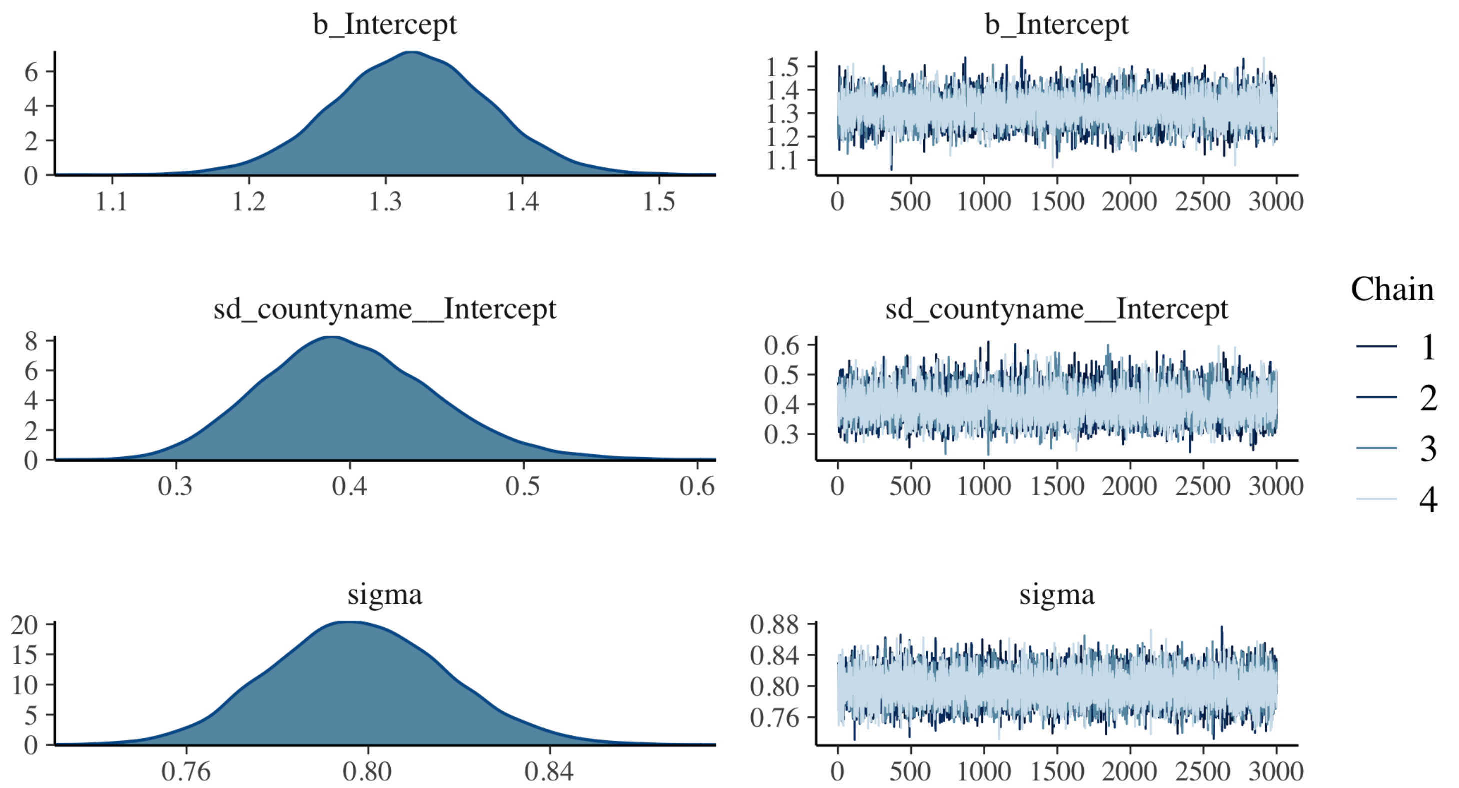
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_radon ~ (1 | countyname)
##    Data: Radon
##
## REML criterion at convergence: 2259.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4661 -0.5734  0.0441  0.6432  3.3516
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  countyname (Intercept) 0.09581  0.3095
##  Residual               0.63662  0.7979
## Number of obs: 919, groups:  countyname, 85
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  1.31258    0.04891   26.84
```

STA 610L

Radon study again

Quick diagnostics for the Bayesian model.

```
plot(m1)
```



STA 610L

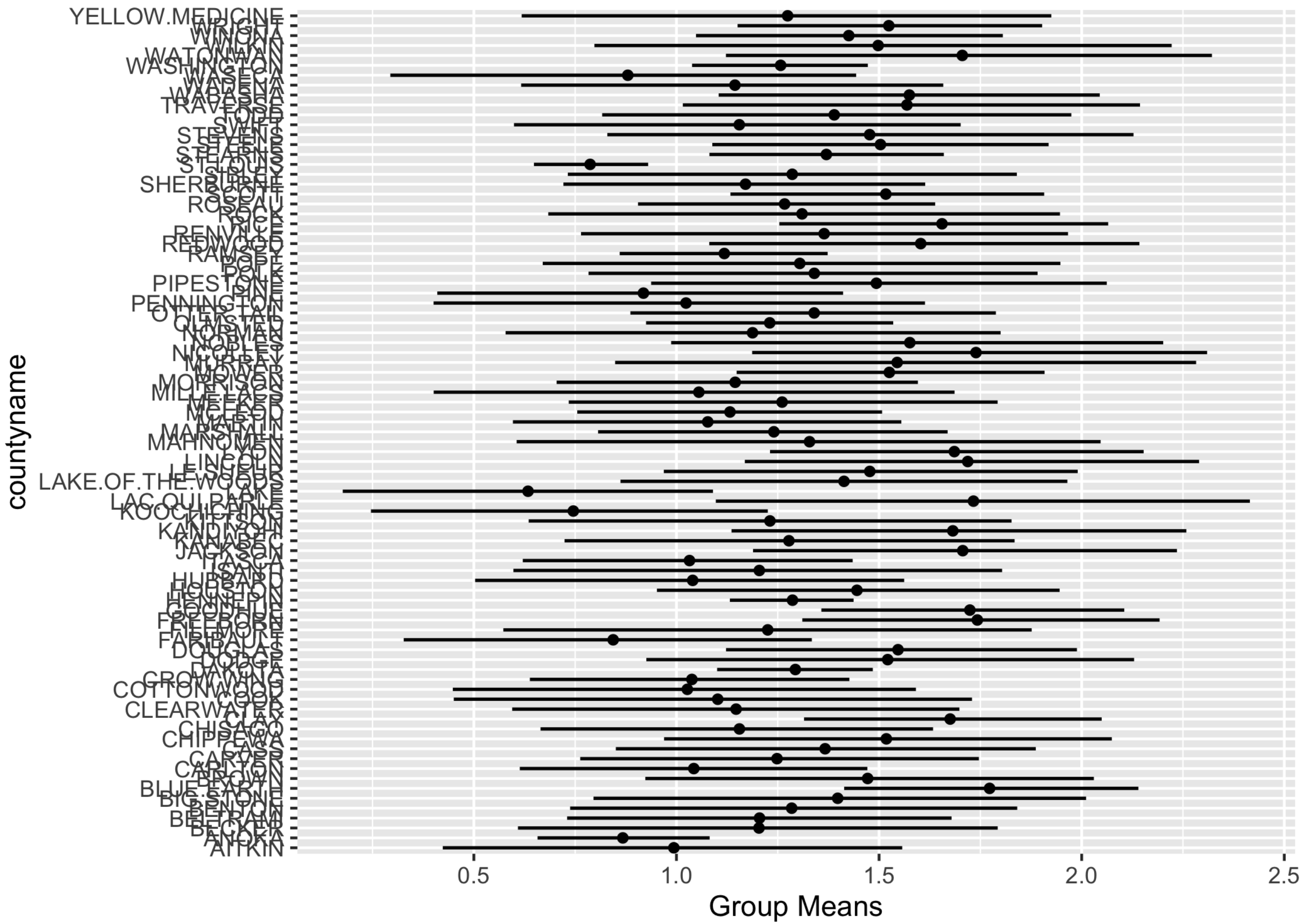
Radon study again

We can plot the group means.

```
m1 %>%
  spread_draws(b_Intercept, r_countyname[countyname,]) %>%
  median_qi(`Group Means` = b_Intercept + r_countyname) %>%
  ggplot(aes(y = countyname, x = `Group Means`, xmin = .lower, xmax = .upper)) +
  geom_pointinterval(orientation = "horizontal")
```

STA 610L

Radon study again

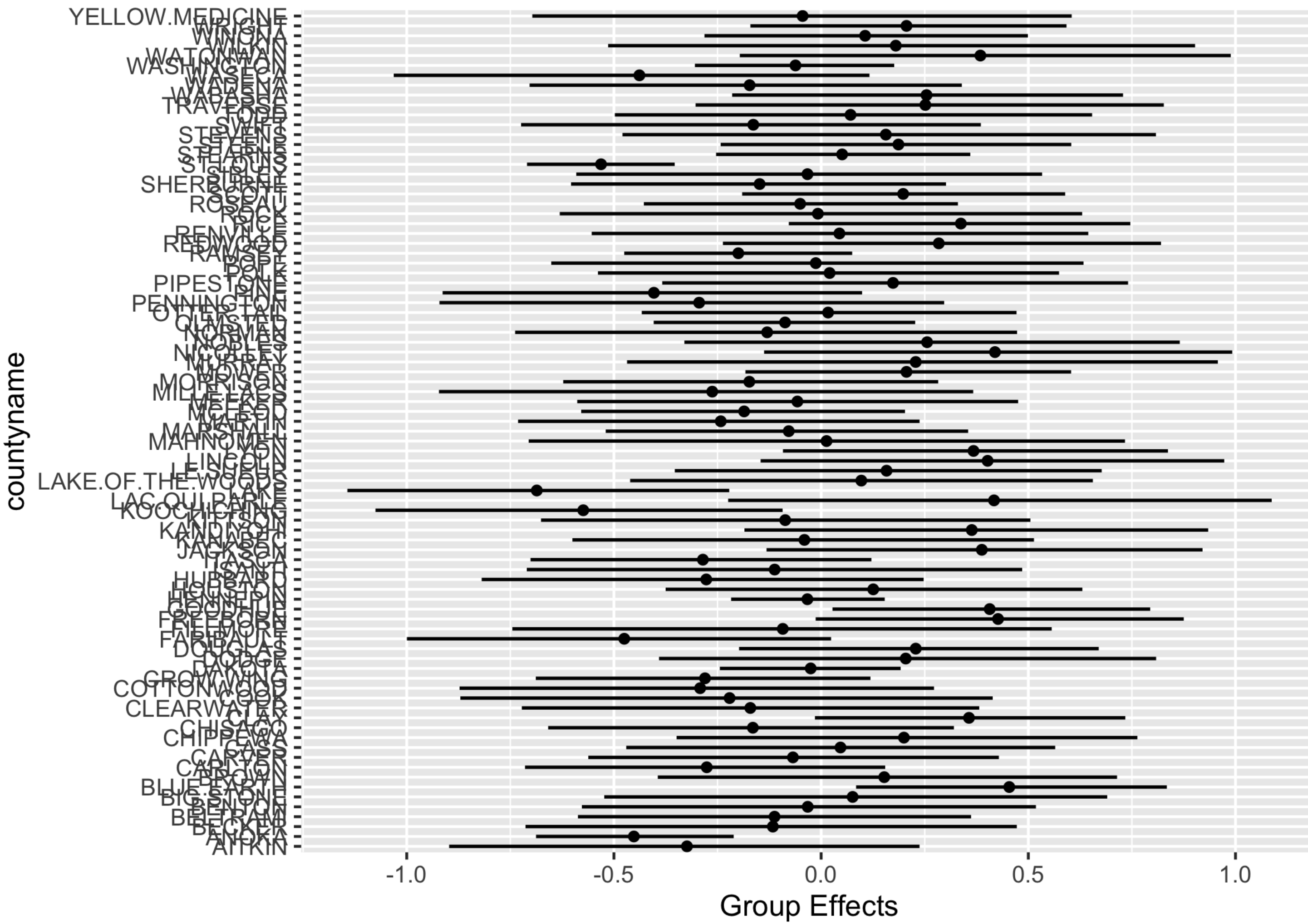


Radon study again

...or just the treatment effects.

```
m1 %>%
  spread_draws(r_countyname[countyname,]) %>%
  median_qi(`Group Effects` = r_countyname) %>%
  ggplot(aes(y = countyname, x = `Group Effects`, xmin = .lower, xmax = .upper)) +
  geom_pointinterval(orientation = "horizontal")
```

Radon study again



Challenge to validity: heterogeneous means and variances

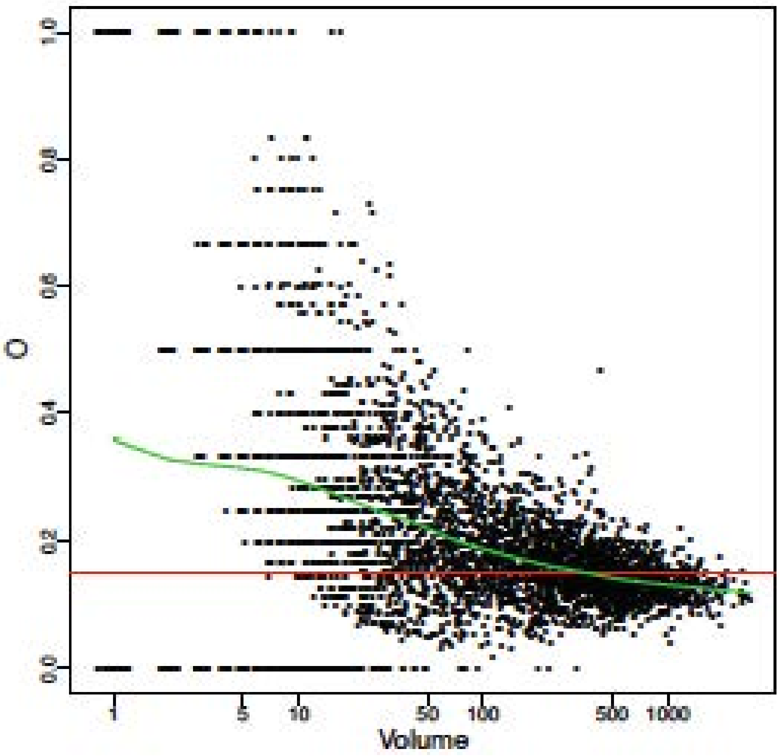
Recall our model again:

$$
\begin{aligned}
y_{ij}|\mu_j, \sigma^2 &\sim \mathcal{N}\left(\mu_j, \sigma^2\right); \quad i = 1, \ldots, n_j \\
\mu_j|\mu, \tau^2 &\sim \mathcal{N}\left(\mu, \tau^2\right); \quad j = 1, \ldots, J, \\
\mu &\sim \mathcal{N}\left(\mu_0, \gamma_0^2\right), \\
\tau^2 &\sim \mathcal{IG}\left(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2}\right), \\
\sigma^2 &\sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right).
\end{aligned}
$$

While we are indeed sharing information across groups, we only do so via the group-specific means.

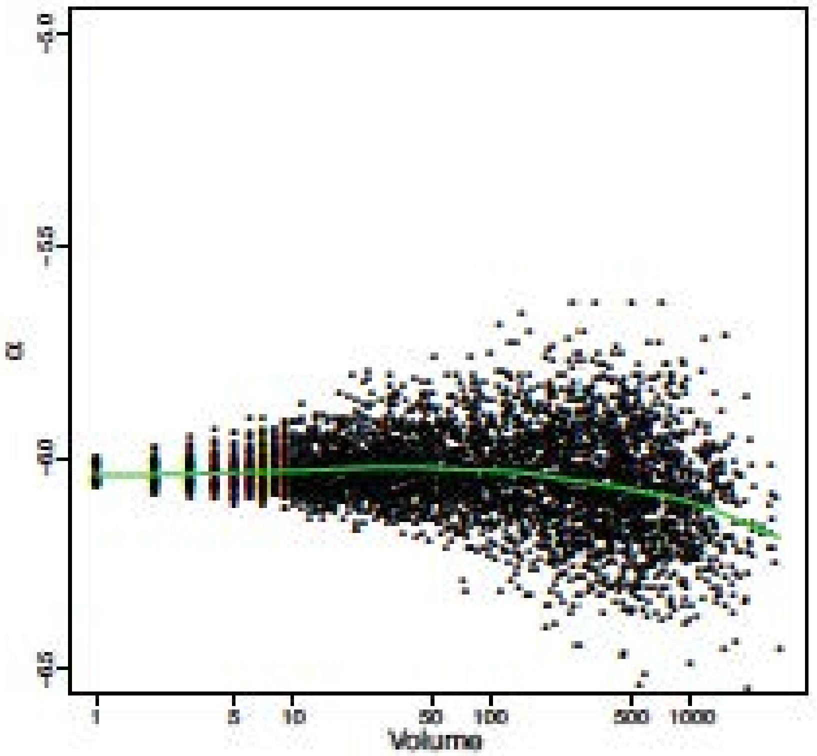
While many people feel that shrinkage can "do no harm", it can be quite detrimental when the shrinkage target is not correctly specified.

Mortality by volume



STA 610L

Estimated random intercepts by volume



Group-specific variances

How might we specify a model to avoid such problems? We could introduce predictors to model group means and or group variances. For example,

$$\alpha_j \sim N(\mu_j(z), \tau_j^2(z))$$

Another potential challenge is that the variance of the response may not be the same for each group anyway. This could be due to a variety of factors.

One potential remedy for this issue is to allow the error variance to differ across groups. A natural extension is

$$\begin{aligned} y_{ij}|\mu_j, \sigma^2 &\sim \mathcal{N}\left(\mu_j, \sigma_j^2\right); \quad i = 1, \ldots, n_j \\ \mu_j|\mu, \tau^2 &\sim \mathcal{N}\left(\mu, \tau^2\right); \quad j = 1, \ldots, J, \\ \sigma_1^2, \ldots, \sigma_J^2|\nu_0, \sigma_0^2 &\sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right). \end{aligned}$$

Posterior inference

- The full posterior is now:

$$
\begin{aligned}
\pi(\mu_1,\ldots,\mu_J,\sigma_1^2,\ldots,\sigma_J^2,\mu,\tau^2,\nu_0,\sigma_0^2|Y) &\propto p(y|\mu_1,\ldots,\mu_J,\sigma_1^2,\ldots,\sigma_J^2,\mu,\tau^2,\nu_0,\sigma_0^2)\\
&\quad\times p(\mu_1,\ldots,\mu_J|\sigma_1^2,\ldots,\sigma_J^2,\mu,\tau^2,\nu_0,\sigma_0^2)\\
&\quad\times p(\sigma_1^2,\ldots,\sigma_J^2|\mu,\tau^2,\nu_0,\sigma_0^2)\\
&\quad\times \pi(\mu,\tau^2,\nu_0,\sigma_0^2)\\
&= p(y|\mu_1,\ldots,\mu_J,\sigma_1^2,\ldots,\sigma_J^2)\\
&\quad\times p(\mu_1,\ldots,\mu_J|\mu,\tau^2)\\
&\quad\times p(\sigma_1^2,\ldots,\sigma_J^2|\nu_0,\sigma_0^2)\\
&\quad\times \pi(\mu)\cdot\pi(\tau^2)\cdot\pi(\nu_0)\cdot\pi(\sigma_0^2)\\
&= \left\{\prod_{j=1}^{J}\prod_{i=1}^{n_j} p(y_{ij}|\mu_j,\sigma_j^2)\right\}\\
&\quad\times\left\{\prod_{j=1}^{J} p(\mu_j|\mu,\tau^2)\right\}\\
&\quad\times\left\{\prod_{j=1}^{J} p(\sigma_j^2|\nu_0,\sigma_0^2)\right\}\\
&\quad\times \pi(\mu)\cdot\pi(\tau^2)\cdot\pi(\nu_0)\cdot\pi(\sigma_0^2)
\end{aligned}
$$

STA 610L

Full conditionals

- Notice that this new factorization won't affect the full conditionals for μ and τ^2 from before, since those have nothing to do with all the new σ_j^2's.

- That is,

$$\pi(\mu|\cdots\cdots) = \mathcal{N}\left(\mu_n, \gamma_n^2\right) \quad \text{where}$$

$$\gamma_n^2 = \frac{1}{\dfrac{J}{\tau^2} + \dfrac{1}{\gamma_0^2}}; \qquad \mu_n = \gamma_n^2 \left[\frac{J}{\tau^2}\bar{\theta} + \frac{1}{\gamma_0^2}\mu_0\right],$$

and

$$\pi(\tau^2|\cdots\cdots) = \mathcal{IG}\left(\frac{\eta_n}{2}, \frac{\eta_n \tau_n^2}{2}\right) \quad \text{where}$$

$$\eta_n = \eta_0 + J; \qquad \tau_n^2 = \frac{1}{\eta_n}\left[\eta_0\tau_0^2 + \sum_{j=1}^{J}(\mu_j - \mu)^2\right].$$

STA 610L

Full conditionals

- The full conditional for each μ_j, we have

$$\pi(\mu_j|\mu_{-j},\mu,\sigma_1^2,\ldots,\sigma_J^2,\tau^2,Y) \propto \left\{\prod_{i=1}^{n_j} p(y_{ij}|\mu_j,\sigma_j^2)\right\} \cdot p(\mu_j|\mu,\tau^2)$$

with the only change from before being σ_j^2.

- That is, those terms still include a normal density for μ_j multiplied by a product of normals in which μ_j is the mean, again mirroring the previous case, so you can show that

$$\pi(\mu_j|\mu_{-j},\mu,\sigma_1^2,\ldots,\sigma_J^2,\tau^2,Y) = \mathcal{N}\left(\mu_j^\star,\tau_j^\star\right) \quad \text{where}$$

$$\tau_j^\star = \frac{1}{\frac{n_j}{\sigma_j^2}+\frac{1}{\tau^2}}; \qquad \mu_j^\star = \tau_j^\star\left[\frac{n_j}{\sigma_j^2}\bar{y}_j + \frac{1}{\tau^2}\mu\right]$$

STA 610L

How about within-group variances?

- Before we get to the choice of the priors for ν_0 and σ_0^2, we have enough to derive the full conditional for each σ_j^2. This actually takes a similar form to what we had before we indexed by j, that is,

$$\pi(\sigma_j^2|\sigma_{-j}^2, \mu_1, \ldots, \mu_J, \mu, \tau^2, \nu_0, \sigma_0^2, Y) \propto \left\{\prod_{i=1}^{n_j} p(y_{ij}|\mu_j, \sigma_j^2)\right\} \cdot \pi(\sigma_j^2|\nu_0, \sigma_0^2)$$

- This still looks like what we had before, that is, products of normals and one inverse-gamma, so that

$$\pi(\sigma_j^2|\sigma_{-j}^2, \mu_1, \ldots, \mu_J, \mu, \tau^2, \nu_0, \sigma_0^2, Y) = \mathcal{IG}\left(\frac{\nu_j^\star}{2}, \frac{\nu_j^\star \sigma_j^{2(\star)}}{2}\right) \quad \text{where}$$

$$\nu_j^\star = \nu_0 + n_j; \qquad \sigma_j^{2(\star)} = \frac{1}{\nu_j^\star}\left[\nu_0\sigma_0^2 + \sum_{i=1}^{n_j}(y_{ij} - \mu_j)^2\right].$$

STA 610L

Remaining hyper-priors

- Now we can get back to priors for ν_0 and σ_0^2. We know that a semi-conjugate prior for σ_0^2 is a gamma distribution. That is, if we set

$$\pi(\sigma_0^2) = \mathcal{G}a\left(a, b\right),$$

then,

$$\begin{aligned}\pi(\sigma_0^2|\mu_1,\ldots,\mu_J,\sigma_1^2,\ldots,\sigma_J^2,\mu,\tau^2,\nu_0,Y) &\propto \left\{\prod_{j=1}^{J} p(\sigma_j^2|\nu_0,\sigma_0^2)\right\} \cdot \pi(\sigma_0^2)\\ &\propto \mathcal{IG}\left(\sigma_j^2; \frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right) \cdot \mathcal{G}a\left(\sigma_0^2; a, b\right)\end{aligned}$$

- Recall that
  - $\mathcal{G}a(y; a, b) \equiv \dfrac{b^a}{\Gamma(a)} y^{a-1} e^{-by}$, and
  - $\mathcal{IG}(y; a, b) \equiv \dfrac{b^a}{\Gamma(a)} y^{-(a+1)} e^{-\dfrac{b}{y}}$.

STA 610L

Remaining hyper-priors

- So $\pi(\sigma_0^2 | \mu_1, \ldots, \mu_J, \sigma_1^2, \ldots, \sigma_J^2, \mu, \tau^2, \nu_0, Y)$

$$
\begin{aligned}
&\propto \left\{ \prod_{j=1}^{J} p(\sigma_j^2 | \nu_0, \sigma_0^2) \right\} \cdot \pi(\sigma_0^2) \\
&\propto \prod_{j=1}^{J} \mathcal{IG}\left(\sigma_j^2; \frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right) \cdot \mathcal{G}a\left(\sigma_0^2; a, b\right) \\
&= \left[\prod_{j=1}^{J} \frac{\left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\left(\frac{\nu_0}{2}\right)}}{\Gamma\left(\frac{\nu_0}{2}\right)} (\sigma_j^2)^{-\left(\frac{\nu_0}{2}+1\right)} e^{-\frac{\nu_0 \sigma_0^2}{2(\sigma_j^2)}} \right] \cdot \left[\frac{b^a}{\Gamma(a)} (\sigma_0^2)^{a-1} e^{-b\sigma_0^2} \right] \\
&\propto \left[\prod_{j=1}^{J} (\sigma_0^2)^{\left(\frac{\nu_0}{2}\right)} e^{-\frac{\nu_0 \sigma_0^2}{2(\sigma_j^2)}} \right] \cdot \left[(\sigma_0^2)^{a-1} e^{-b\sigma_0^2} \right] \\
&\propto \left[(\sigma_0^2)^{\left(\frac{J\nu_0}{2}\right)} e^{-\sigma_0^2 \left[\frac{\nu_0}{2} \sum_{j=1}^{J} \frac{1}{\sigma_j^2} \right]} \right] \cdot \left[(\sigma_0^2)^{a-1} e^{-b\sigma_0^2} \right]
\end{aligned}
$$

STA 610L

Remaining hyper-priors

- That is, the full conditional is

$$\begin{aligned}\pi(\sigma_0^2|\cdots\cdots) &\propto \left[(\sigma_0^2)^{\left(\frac{J\nu_0}{2}\right)} e^{-\sigma_0^2\left[\frac{\nu_0}{2}\sum_{j=1}^{J}\frac{1}{\sigma_j^2}\right]}\right]\cdot\left[(\sigma_0^2)^{a-1}e^{-b\sigma_0^2}\right]\\ &\propto \left[(\sigma_0^2)^{\left(a+\frac{J\nu_0}{2}-1\right)} e^{-\sigma_0^2\left[b+\frac{\nu_0}{2}\sum_{j=1}^{J}\frac{1}{\sigma_j^2}\right]}\right]\\ &\equiv \mathcal{G}a\left(\sigma_0^2; a_n, b_n\right),\end{aligned}$$

where

$$a_n = a + \frac{J\nu_0}{2}; \quad b_n = b + \frac{\nu_0}{2}\sum_{j=1}^{J}\frac{1}{\sigma_j^2}.$$

STA 610L

Remaining hyper-priors

- OK that leaves us with one parameter to go, i.e., ν_0. Turns out there is no simple conjugate/semi-conjugate prior for ν_0.

- Given that we know how to do Metropolis/Metropolis-Hastings, we actually have many options here, but to keep this simple, let's follow the same path as what you (hopefully) did for this model in STA 360/601/602.

- That is, restrict ν_0 to be an integer (which makes sense when we think of it as being degrees of freedom, which also means it cannot be zero). With the restriction, we need a discrete distribution as the prior with support on $\nu_0 = 1, 2, 3, \ldots$.

- A popular choice is the geometric distribution with pmf $p(\nu_0) = (1-p)^{\nu_0 - 1} p$.

- However, we can rewrite the kernel as $\pi(\nu_0) \propto e^{-\alpha \nu_0}$. How did we get here from the geometric pmf and what is α?

STA 610L

Final full conditional

- With this prior, $\pi(\nu_0 | \mu_1, \ldots, \mu_J, \sigma_1^2, \ldots, \sigma_J^2, \mu, \tau^2, \sigma_0^2, Y)$

$$\propto \left\{ \prod_{j=1}^{J} p(\sigma_j^2 | \nu_0, \sigma_0^2) \right\} \cdot \pi(\nu_0)$$

$$\propto \prod_{j=1}^{J} \mathcal{IG}\left(\sigma_j^2; \frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2} \right) \cdot e^{-\alpha \nu_0}$$

$$= \left[\prod_{j=1}^{J} \frac{\left(\frac{\nu_0 \sigma_0^2}{2} \right)^{\left(\frac{\nu_0}{2} \right)}}{\Gamma\left(\frac{\nu_0}{2} \right)} \left(\sigma_j^2 \right)^{-\left(\frac{\nu_0}{2} + 1 \right)} e^{-\frac{\nu_0 \sigma_0^2}{2(\sigma_j^2)}} \right] \cdot e^{-\alpha \nu_0}$$

$$\propto \left[\left(\frac{\left(\frac{\nu_0 \sigma_0^2}{2} \right)^{\left(\frac{\nu_0}{2} \right)}}{\Gamma\left(\frac{\nu_0}{2} \right)} \right)^{J} \cdot \left(\prod_{j=1}^{J} \frac{1}{\sigma_j^2} \right)^{\left(\frac{\nu_0}{2} + 1 \right)} \cdot e^{-\nu_0 \left[\frac{\sigma_0^2}{2} \sum_{j=1}^{J} \frac{1}{\sigma_j^2} \right]} \right] \cdot e^{-\alpha \nu_0}$$

STA 610L

Final full conditional

- That is, the full conditional is

$$\pi(\nu_0|\cdots\cdots) \propto \left[\left(\frac{\left(\frac{\nu_0\sigma_0^2}{2}\right)^{\left(\frac{\nu_0}{2}\right)}}{\Gamma\left(\frac{\nu_0}{2}\right)}\right)^J \cdot \left(\prod_{j=1}^J \frac{1}{\sigma_j^2}\right)^{\left(\frac{\nu_0}{2}+1\right)} \cdot e^{-\nu_0\left[\alpha + \frac{\sigma_0^2}{2}\sum_{j=1}^J \frac{1}{\sigma_j^2}\right]}\right],$$

  which is not a known kernel and is thus unnormalized (i.e., does not integrate to 1 in its current form).

- While this looks like a lot, it is relatively easy to compute in R, for a grid of ν_0 values.

- Technically, the support is $\nu_0 = 1, 2, 3, \ldots$,but we can compute the unnormalized distribution across say $\nu_0 = 1, 2, 3, \ldots, K$ for some large K, re-normalize, and then sample.

STA 610L

Final full conditional

- One more thing, computing these probabilities on the raw scale can be problematic particularly because of the product inside. Good idea to transform to the log scale instead.
- That is,

$$\pi(\nu_0|\cdots\cdots) \propto \left[\left(\frac{\left(\frac{\nu_0\sigma_0^2}{2}\right)^{\left(\frac{\nu_0}{2}\right)}}{\Gamma\left(\frac{\nu_0}{2}\right)} \right)^J \cdot \left(\prod_{j=1}^J \frac{1}{\sigma_j^2} \right)^{\left(\frac{\nu_0}{2}-1\right)} \cdot e^{-\nu_0\left[\alpha + \frac{\sigma_0^2}{2}\sum_{j=1}^J \frac{1}{\sigma_j^2}\right]} \right]$$

$$\Rightarrow \ln\pi(\nu_0|\cdots\cdots) \propto \left(\frac{J\nu_0}{2}\right)\ln\left(\frac{\nu_0\sigma_0^2}{2}\right) - J\ln\left[\Gamma\left(\frac{\nu_0}{2}\right)\right] + \left(\frac{\nu_0}{2}+1\right)\left(\sum_{j=1}^J \ln\left[\frac{1}{\sigma_j^2}\right]\right) - \nu_0\left[\alpha + \frac{\sigma_0^2}{2}\sum_{j=1}^J \frac{1}{\sigma_j^2}\right]$$

STA 610L

Full model

As a recap, the final model is:

$$y_{ij}|\mu_j,\sigma_j^2 \sim \mathcal{N}\left(\mu_j,\sigma_j^2\right); \quad i=1,\ldots,n_j; \quad j=1,\ldots,J$$

$$\mu_j|\mu,\tau^2 \sim \mathcal{N}\left(\mu,\tau^2\right); \quad j=1,\ldots,J$$

$$\sigma_1^2,\ldots,\sigma_J^2|\nu_0,\sigma_0^2 \sim \mathcal{IG}\left(\frac{\nu_0}{2},\frac{\nu_0\sigma_0^2}{2}\right); \quad j=1,\ldots,J$$

$$\mu \sim \mathcal{N}\left(\mu_0,\gamma_0^2\right)$$

$$\tau^2 \sim \mathcal{IG}\left(\frac{\eta_0}{2},\frac{\eta_0\tau_0^2}{2}\right).$$

$$\pi(\nu_0) \propto e^{-\alpha\nu_0}$$

$$\sigma_0^2 \sim \mathcal{G}a\left(a,b\right).$$

STA 610L

Gibbs sampler

```
#Data summaries
J <- #number of groups
ybar <- #vector of the group sample means
s_j_sq <- #vector of the group sample variances
n <- #vector of the number of observations in each group


#Hyperparameters for the priors
mu_0 <- ...
gamma_0_sq <- ...
eta_0 <- ...
tau_0_sq <- ...
alpha <- ...
a <- ...
b <- ...


#Grid values for sampling nu_0_grid
nu_0_grid <- 1:5000


#Initial values for Gibbs sampler
theta <- ybar #theta vector for all the mu_j's
sigma_sq <- s_j_sq
mu <- mean(theta)
tau_sq <- var(theta)
nu_0 <- 1
sigma_0_sq <- 100
```

STA 610L

Gibbs sampler

```r
#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

#Set null matrices to save samples
SIGMA_SQ <- THETA <- matrix(nrow=n_iter, ncol=J)
OTHER_PAR <- matrix(nrow=n_iter, ncol=4)

#Now, to the Gibbs sampler
for(s in 1:(n_iter+burn_in)){

  #update the theta vector (all the mu_j's)
  tau_j_star <- 1/(n/sigma_sq + 1/tau_sq)
  mu_j_star <- tau_j_star*(ybar*n/sigma_sq + mu/tau_sq)
  theta <- rnorm(J,mu_j_star,sqrt(tau_j_star))

  #update the sigma_sq vector (all the sigma_sq_j's)
  nu_j_star <- nu_0 + n
  theta_long <- rep(theta,n)
  nu_j_star_sigma_j_sq_star <-
    nu_0*sigma_0_sq + c(by((Y[,"mathscore"] - theta_long)^2,Y[,"school"],sum))
  sigma_sq <- 1/rgamma(J,(nu_j_star/2),(nu_j_star_sigma_j_sq_star/2))

  #update mu
  gamma_n_sq <- 1/(J/tau_sq + 1/gamma_0_sq)
  mu_n <- gamma_n_sq*(J*mean(theta)/tau_sq + mu_0/gamma_0_sq)
  mu <- rnorm(1,mu_n,sqrt(gamma_n_sq))
```

STA 610L

Gibbs sampler

```
  #update tau_sq
  eta_n <- eta_0 + J
  eta_n_tau_n_sq <- eta_0*tau_0_sq + sum((theta-mu)^2)
  tau_sq <- 1/rgamma(1,eta_n/2,eta_n_tau_n_sq/2)

  #update sigma_0_sq
  sigma_0_sq <- rgamma(1,(a + J*nu_0/2),(b + nu_0*sum(1/sigma_sq)/2))

  #update nu_0
  log_prob_nu_0 <- (J*nu_0_grid/2)*log(nu_0_grid*sigma_0_sq/2) -
    J*lgamma(nu_0_grid/2) +
    (nu_0_grid/2+1)*sum(log(1/sigma_sq)) -
    nu_0_grid*(alpha + sigma_0_sq*sum(1/sigma_sq)/2)
  nu_0 <- sample(nu_0_grid,1, prob = exp(log_prob_nu_0 - max(log_prob_nu_0)) )
  #this last step substracts the maximum logarithm from all logs
  #it is a neat trick that throws away all results that are so negative
  #they will screw up the exponential
  #note that the sample function will renormalize the probabilities internally


  #save results only past burn-in
  if(s > burn_in){
    THETA[(s-burn_in),] <- theta
    SIGMA_SQ[(s-burn_in),] <- sigma_sq
    OTHER_PAR[(s-burn_in),] <- c(mu,tau_sq,sigma_0_sq,nu_0)
  }
}
colnames(OTHER_PAR) <- c("mu","tau_sq","sigma_0_sq","nu_0")
```

What's next?

Move on to the readings for the next module!

STA 610L