

hw2

Will Tirone

Q1)

To modify the Gibb's sampler, we just had to update the full conditionals for τ^2 and μ . I have attached handwritten work for this and updated the Gibb's sampler below.

```
gibbs = function(data, target, group, iter,
                  mu_0, kappa_0, eta_0, tau_0_sq,
                  alpha, a, b) {

  # need !! syntax to pass in a group / target vars
  grouped = data |>
    group_by(!!sym(group))

  J = data |>
    select(!!sym(group)) |>
    distinct() |>
    summarise(c = n()) |>
    pull()

  ybar <- grouped |>
    summarise(mean(!!sym(target))) |>
    pull()

  # had some NA values, so just replacing these with a value
  s_j_sq <- grouped |>
    summarise(v = var(!!sym(target))) |>
    replace_na(list(v = 4)) |>
    pull()

  n <- grouped |>
    summarise(n()) |>
```

```

pull()

# Hyperparameters for the priors
mu_0 <- mu_0

# NEW
kappa_0 = kappa_0

eta_0 <- eta_0
tau_0_sq <- tau_0_sq
alpha <- alpha
a <- a
b <- 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

# First, set the number of iterations and burn-in, then set the seed
n_iter <- iter
burn_in <- 0.3 * n_iter

# 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((radon[, "radon"] - theta_long)^2, radon[, "countynome"], sum))
sigma_sq <- 1 / rgamma(J, (nu_j_star / 2), (nu_j_star_sigma_j_sq_star / 2))

# UPDATED BASED ON NEW PRIOR
gamma_n_sq <- tau_sq / (kappa_0 + J)
mu_n = (mu_0 * kappa_0 + J * mean(theta)) / (kappa_0 + J)
mu <- rnorm(1, mu_n, sqrt(gamma_n_sq))

# UPDATED BASED ON NEW PRIOR
eta_n <- eta_0 + J + 1 # just added one to change this
eta_n_tau_n_sq <- eta_0 * tau_0_sq + sum((theta - mu)^2) +
  kappa_0 * (mu - mu_0)^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)))

# 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")

colnames(THETA) = radon |> distinct(countynome) |> pull()
colnames(SIGMA_SQ) = radon |> distinct(countynome) |> pull()
SIGMA_SQ = data.frame(SIGMA_SQ)

```

```

    THETA = data.frame(THETA)

    output = list(THETA, SIGMA_SQ, OTHER_PAR)

    return(output)
}

```

Now running our function, I have done my best to choose weak priors:

If κ_0 is our prior sample size, setting it to five will not indicate we have strong certainty about information from the prior on μ . From a quick Google search, the average radon level in homes is 1.3, so we'll choose that as μ_0 . Let's say we want a gamma distribution with a long tail, so we'll set $a = b = 50$.

For some different values, I'll set very strong priors just to visualize what happens.

```

# weak priors
sampler_out = gibbs(radon, 'radon', 'countyname',
                    iter=2000, mu_0=1.3, kappa_0 = 5,
                    eta_0 = 10, tau_0_sq = 10, alpha=4,
                    a=50, b=50)

# strong priors
sampler2_out = gibbs(radon, 'radon', 'countyname',
                    iter=2000, mu_0=10000, eta_0=100000,
                    kappa_0 = 1000000, tau_0_sq=100000, alpha=400,
                    a=1, b=1)

```

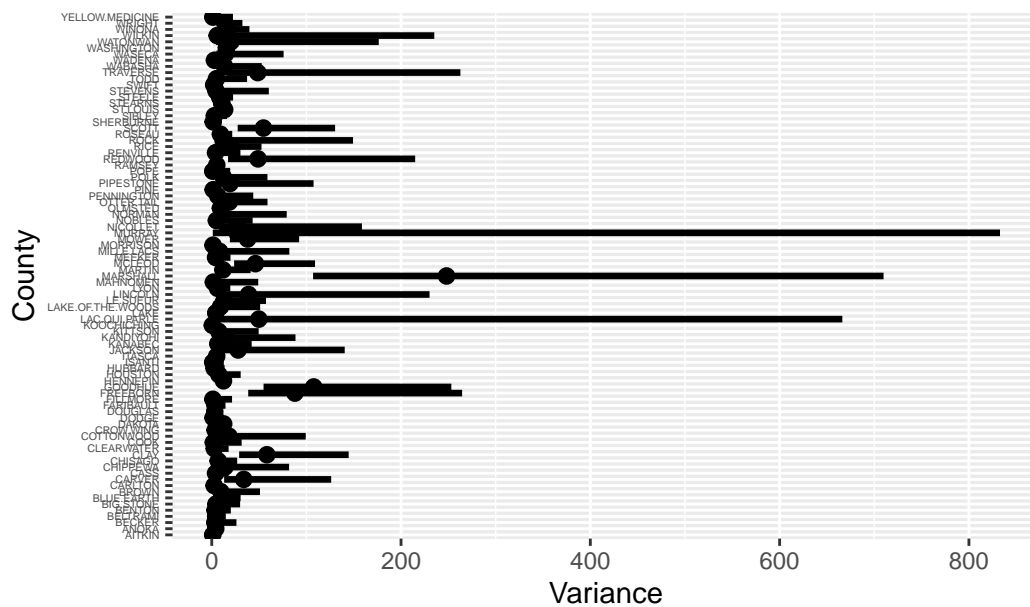
Now, we can plot the error variances across the counties. With our weak prior specification, we can still see fairly convincing evidence that the ranges are not the same across counties. Most counties have a very tight interval around 0, but some have very wide intervals, indicating the error variances are not the same across counties.

```

sampler_out[[2]] |>
  gather() |>
  rename(County = key,
         Variance = value) |>
  group_by(County) |>
  median_qi(.width = 0.95) |>
  ggplot(aes(y = County, x = Variance, xmin = .lower, xmax = .upper)) +
  geom_pointinterval() +
  theme(axis.text.y = element_text(hjust = 1, size = 4)) +
  ggtitle("Radon Variances With Weak Priors")

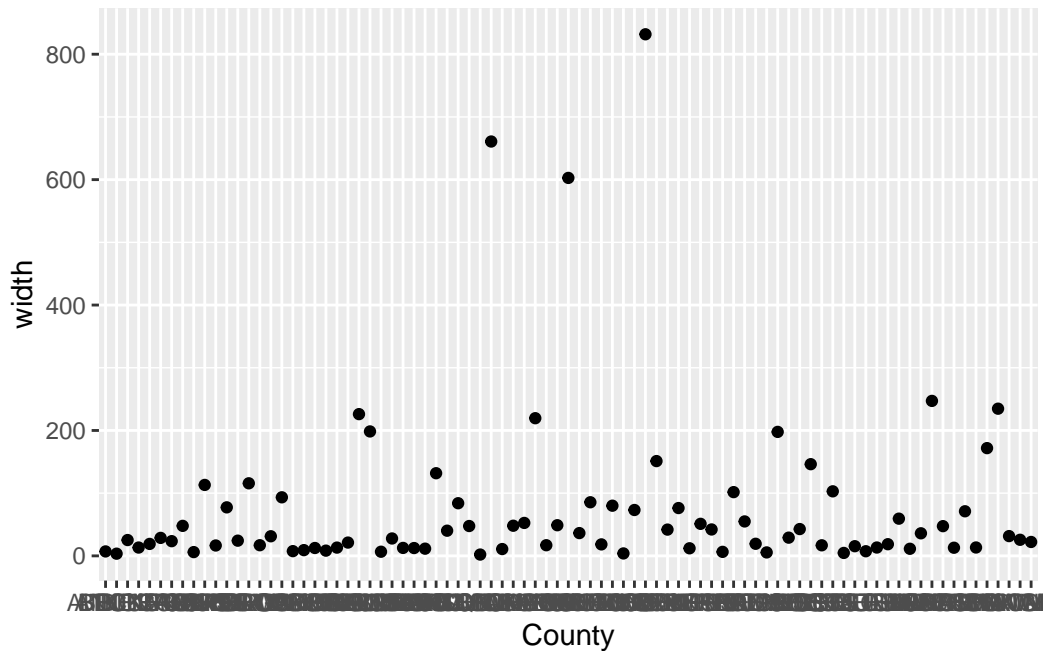
```

Radon Variances With Weak Priors



Visualized slightly differently, we can look at the width of the county's intervals, and see that three of them have enormous widths, a handful have fairly wide intervals of about 250, and many have a width of almost zero.

```
sampler_out[[2]] |>
  gather() |>
  rename(County = key,
         Variance = value) |>
  group_by(County) |>
  median_qi(.width = 0.95) |>
  mutate(width = .upper - .lower) |>
  select(County, width) |>
  arrange(desc(width)) |>
  ggplot(aes(x=County, y=width)) +
  geom_point()
```



With a $\mu_0 = 10,000$ and a huge “prior sample size” $\kappa_0 = 1,000,000$ it seems like this would provide evidence that the rest of the country is an unsurvivable radioactive wasteland, but it seems Minnesota is mostly okay. Most variances are 0, though some are slightly higher and with very wide intervals.

```
sampler2_out[[2]] |>
  gather() |>
  rename(County = key,
         Variance = value) |>
  group_by(County) |>
  median_qi(.width = 0.95) |>
  ggplot(aes(y = County, x = Variance, xmin = .lower, xmax = .upper)) +
  geom_pointinterval() +
  theme(axis.text.y = element_text(hjust = 1, size = 4)) +
  ggtitle("Radon Variances With Strong Priors")
```


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

Because we don't have the scale of grades on the test, it's hard to set actual hyperparameters. However, the power of the HNM here is that some universities have a small number of test takers and other have a large number, so we want to borrow information across groups to create a more accurate mean. I think the primary assumption here is that variance does not differ across universities.

From here, we would either write a Gibbs Sampler with the output mean vector as our primary parameter of interest, or we would use brms. Then, with our output draws from the full conditional distribution of μ , we could do what we did in problem 1), simply plot the point estimate (the mean of μ) and calculate a Bayesian credible interval. We could then rank by the mean of μ , but take into account the variability.

The primary difference between this approach and ranking purely based on the means is that the means within a university would not quantify variability in any way, at least unless we had the number of test takers within the school. Since the question says *entirely* based on the mean test scores, I'll assume we don't have that information. However, this is an important piece, since a school with 3 test takers might have a huge but unreliable average score, but a school with a lower average school and 10,000 test takers would have a much more reliable estimate. Our efforts to control variance across the schools address this problem.