

Variational Bayes: Unified M1/M3 Analysis

VI-UNIFIED-VBGibbsOnly2.Rmd | Compiled: 2026-01-22 08:46:55

Contents

1	Scenario 1: Conditional on Model Type	3
2	Scenario 2: Conditional on Model Type (M3 only)	7
3	Comparison Between Scenarios (M3 only)	11
4	Sample Size Effects on τ_u (Multi-Configuration Analysis)	11

```
# =====  
# MODEL TYPE SELECTION  
# M1: Linear regression with  $\tau$  and  $\sigma^2$  (residual variance)  
#     - No random effects, no variance component  
  
# M3: Hierarchical model with  $\tau$ ,  $u$  (random effects),  $\sigma^2$   
#     - Demonstrates variance component under-dispersion  
#     - Two scenarios: 30 sparse groups vs 6 rich groups  
model_type <- "M3" # "M1" for linear regression, "M3" for hierarchical  
  
# SHARED PARAMETERS (both scenarios)  
n <- 300 # Total observations  
p <- 4   # Number of fixed effects (beta_0, beta_1, beta_2, beta_3)  
  
# True Parameter Values  
tau_e_true <- 5  
tau_u_true <- if (model_type == "M1") NULL else 0.5 # Lower = more variance between groups  
beta_true  <- c(5, 0.5, 0.5, 0.5)  
  
# Prior Hyperparameters  
alpha_e <- 0.01  
gamma_e <- 0.01  
alpha_u <- 1.0 # Prior for tau_u centred at 0.5  
gamma_u <- 2.0 # Mean = alpha_u / gamma_u = 1.0 / 2.0 = 0.5  
  
# Gibbs Sampler Settings  
run_gibbs <- TRUE # Set FALSE for quick testing, TRUE for full run  
gibbs_iter <- 5000  
gibbs_burnin <- 1000  
  
# Display Settings  
RENDER_FUNCTIONS <- FALSE # TRUE to show function code, FALSE to hide  
  
# SCENARIO 1: Q=5 groups (n=60 per group) - M3 only  
q_s1 <- if (model_type == "M1") 0 else 5
```

```
nq_s1 <- if (model_type == "M1") 0 else 60

# SCENARIO 2: Q=50 groups (n=6 per group) - M3 only
q_s2 <- if (model_type == "M1") 0 else 50
nq_s2 <- if (model_type == "M1") 0 else 6

# =====
```

1 Scenario 1: Conditional on Model Type

```

cat(glue("\n=== Scenario 1: Q={q_s1} groups (n={nq_s1} per group) ===\n"))

## === Scenario 1: Q=5 groups (n=60 per group) ===

X_s1 <- cbind(1, matrix(rnorm(n*(p-1)), nrow=n, ncol=p-1))

if (model_type == "M3") {
  u_true_s1 <- rnorm(q_s1, 0, 1/sqrt(tau_u_true))
  Z_s1 <- table(1:n, rep(1:q_s1, each=nq_s1))
  K_s1 <- diag(q_s1)
  linear_predictor_s1 <- X_s1 %*% beta_true + Z_s1 %*% u_true_s1
} else {
  u_true_s1 <- NULL
  Z_s1 <- matrix(0, nrow=n, ncol=1)
  K_s1 <- matrix(1)
  linear_predictor_s1 <- X_s1 %*% beta_true
}

residuals_true_s1 <- rnorm(n, 0, 1/sqrt(tau_e_true))
y_s1 <- as.vector(linear_predictor_s1 + residuals_true_s1)

scenario_name <- if (model_type == "M3") glue("SCENARIO 1: {q_s1} levels ({nq_s1} obs each)") else "SCENARIO 1: 5 levels (60 obs each)"
cat(glue("\n=== {scenario_name} ===\n"))

## === SCENARIO 1: 5 levels (60 obs each) ===

results_s1 <- run_vb_algorithm(
  X      = X_s1,
  Z      = Z_s1,
  y      = y_s1,
  K      = K_s1,
  p      = p,
  q      = q_s1,
  n      = n,
  alpha_e = alpha_e,
  gamma_e = gamma_e,
  alpha_u = alpha_u,
  gamma_u = gamma_u,
  model_type = model_type
)

## Converged at iteration 8
## Max relative change: 5.82e-06

if (run_gibbs) {
  cat("\nRunning Gibbs sampler...\n")
  gibbs_s1 <- run_gibbs_sampler(
    X      = X_s1,
    Z      = Z_s1,
    y      = y_s1,
    p      = p,
    q      = q_s1,
    n      = n,
    alpha_e = alpha_e,

```

```

    gamma_e = gamma_e,
    alpha_u = alpha_u,
    gamma_u = gamma_u,
    model_type = model_type,
    n_iter = gibbs_iter,
    n_burnin = gibbs_burnin
  )

  cat("Gibbs posterior means:\n")
  cat("beta:", colMeans(gibbs_s1[, 1:p]), "\n")
  cat("tau_e:", mean(gibbs_s1[, "tau_e"]), "\n")
  if (model_type == "M3") {
    cat("tau_u:", mean(gibbs_s1[, "tau_u"]), "\n")
  }

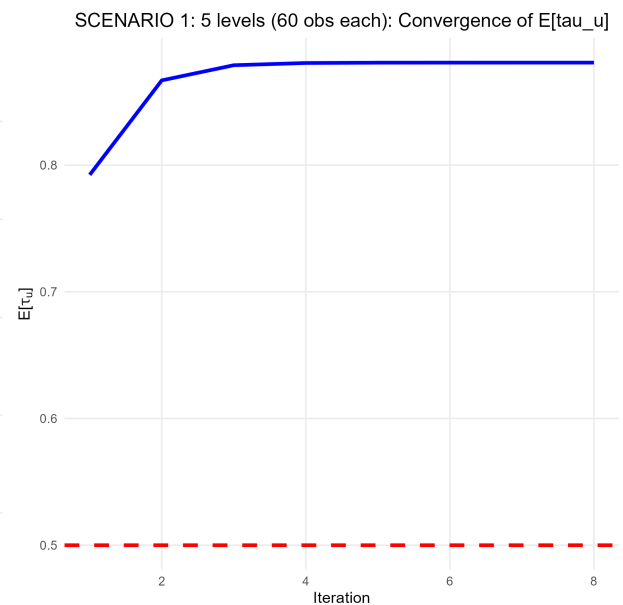
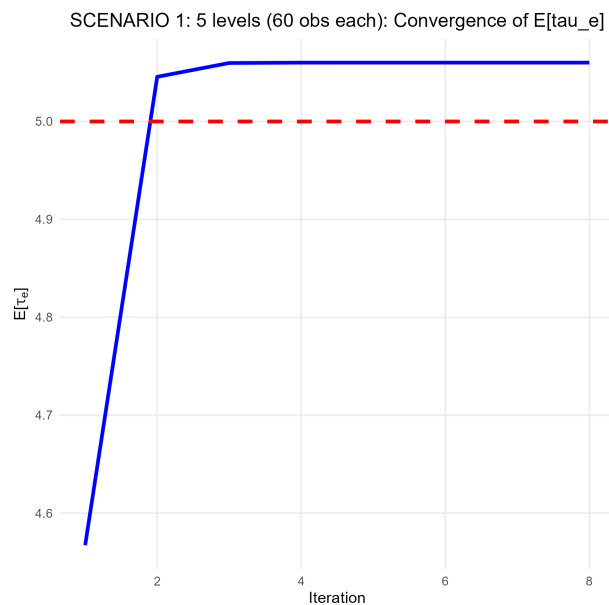
  gibbs_tau_e_s1 <- gibbs_s1[, "tau_e"]
  gibbs_tau_u_s1 <- if (model_type == "M3") gibbs_s1[, "tau_u"] else NULL
} else {
  gibbs_s1 <- NULL
  gibbs_tau_e_s1 <- NULL
  gibbs_tau_u_s1 <- NULL
}

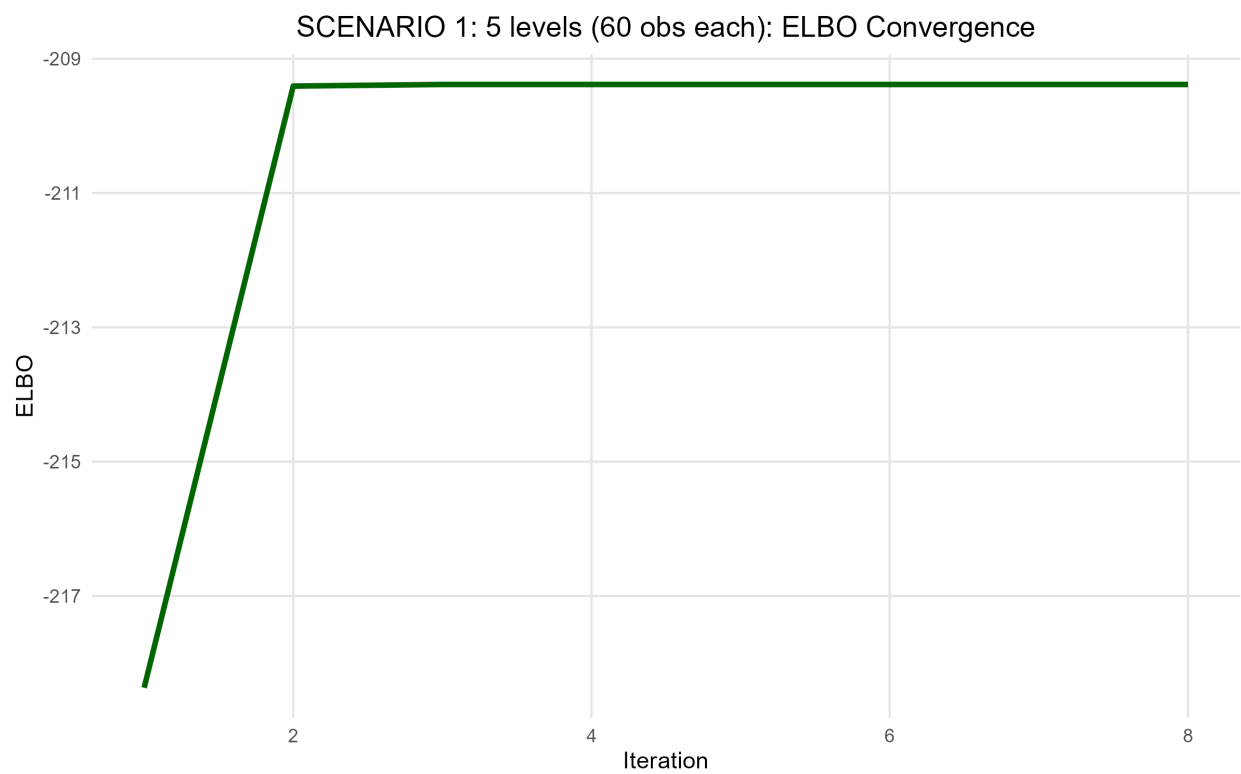
```

```

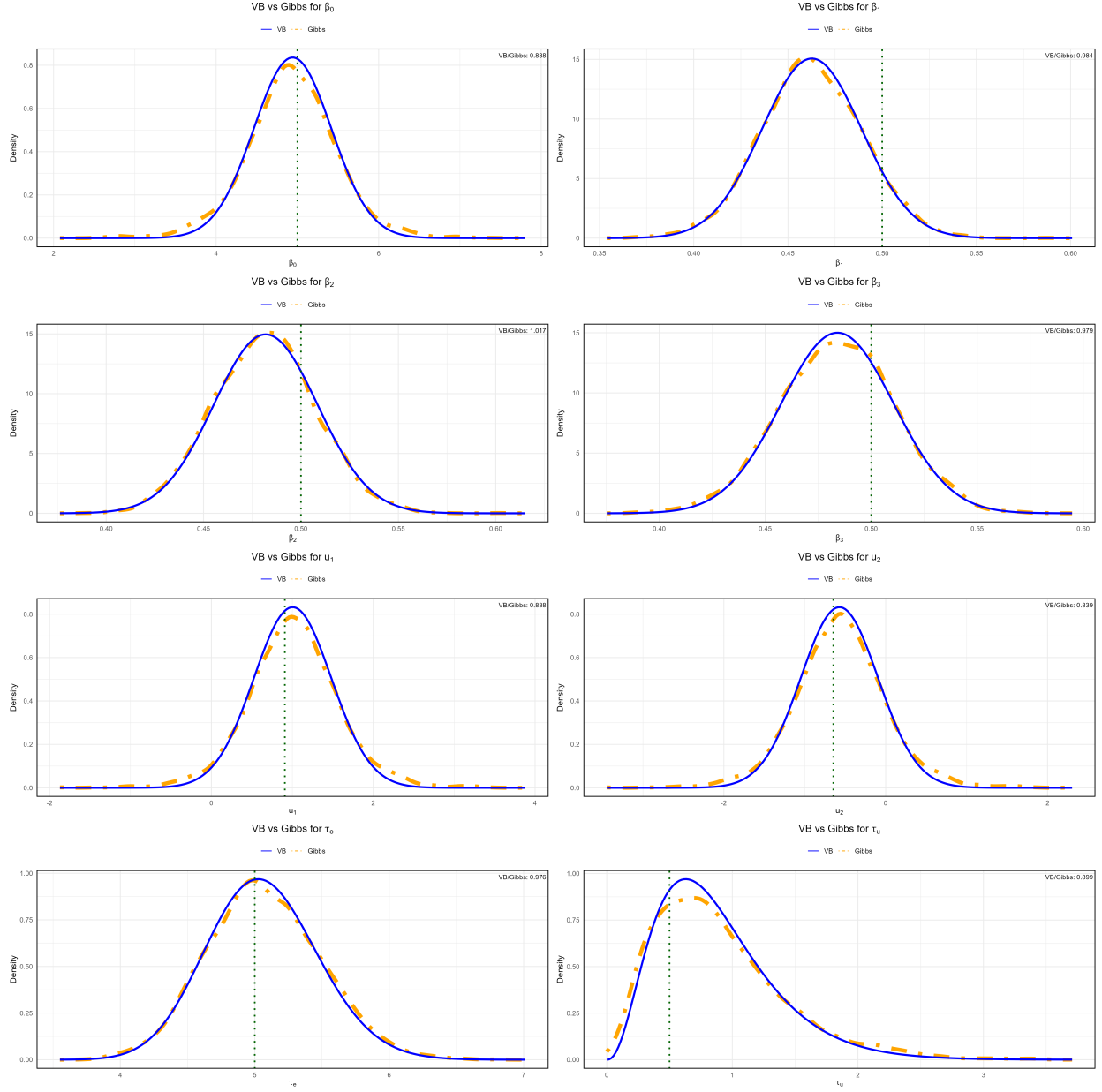
##
## Running Gibbs sampler...
## Gibbs posterior means:
## beta: 4.927609 0.4625412 0.4813218 0.4838531
## tau_e: 5.070511
## tau_u: 0.8942215

```





M3: 8-panel ggplot saved for Scenario 1



2 Scenario 2: Conditional on Model Type (M3 only)

```
if (model_type == "M3") {
  cat(glue("\n=== Scenario 2: Q={q_s2} groups (n={nq_s2} per group) ===\n"))

  u_true_s2 <- rnorm(q_s2, 0, 1/sqrt(tau_u_true))

  X_s2 <- cbind(1, matrix(rnorm(n*(p-1)), nrow=n, ncol=p-1))
  Z_s2 <- table(1:n, rep(1:q_s2, each=nq_s2))
  K_s2 <- diag(q_s2)

  linear_predictor_s2 <- X_s2 %*% beta_true + Z_s2 %*% u_true_s2
  residuals_true_s2 <- rnorm(n, 0, 1/sqrt(tau_e_true))
  y_s2 <- as.vector(linear_predictor_s2 + residuals_true_s2)
} else {
  cat("\nScenario 2 skipped (M3 only)\n")
}
```

```
## === Scenario 2: Q=50 groups (n=6 per group) ===
```

```
results_s2 <- run_vb_algorithm(
  X      = X_s2,
  Z      = Z_s2,
  y      = y_s2,
  K      = K_s2,
  p      = p,
  q      = q_s2,
  n      = n,
  alpha_e = alpha_e,
  gamma_e = gamma_e,
  alpha_u = alpha_u,
  gamma_u = gamma_u,
  model_type = model_type
)
```

```
## Converged at iteration 9
## Max relative change: 5.40e-06
```

```
if (run_gibbs) {
  cat("\nRunning Gibbs sampler...\n")
  gibbs_s2 <- run_gibbs_sampler(
    X      = X_s2,
    Z      = Z_s2,
    y      = y_s2,
    p      = p,
    q      = q_s2,
    n      = n,
    alpha_e = alpha_e,
    gamma_e = gamma_e,
    alpha_u = alpha_u,
    gamma_u = gamma_u,
    model_type = model_type,
    n_iter   = gibbs_iter,
    n_burnin = gibbs_burnin
  )
}
```

```

cat("Gibbs posterior means:\n")
cat("beta:", colMeans(gibbs_s2[, 1:p]), "\n")
cat("tau_e:", mean(gibbs_s2[, "tau_e"]), "\n")
cat("tau_u:", mean(gibbs_s2[, "tau_u"]), "\n")

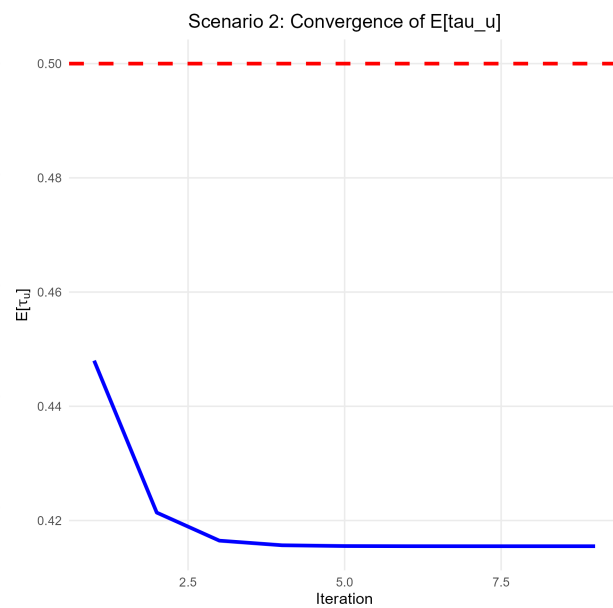
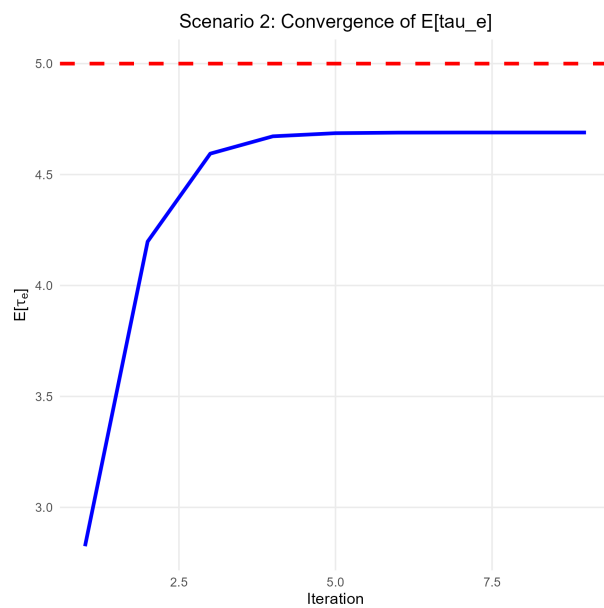
gibbs_tau_e_s2 <- gibbs_s2[, "tau_e"]
gibbs_tau_u_s2 <- gibbs_s2[, "tau_u"]
} else {
  gibbs_s2 <- NULL
  gibbs_tau_e_s2 <- NULL
  gibbs_tau_u_s2 <- NULL
}

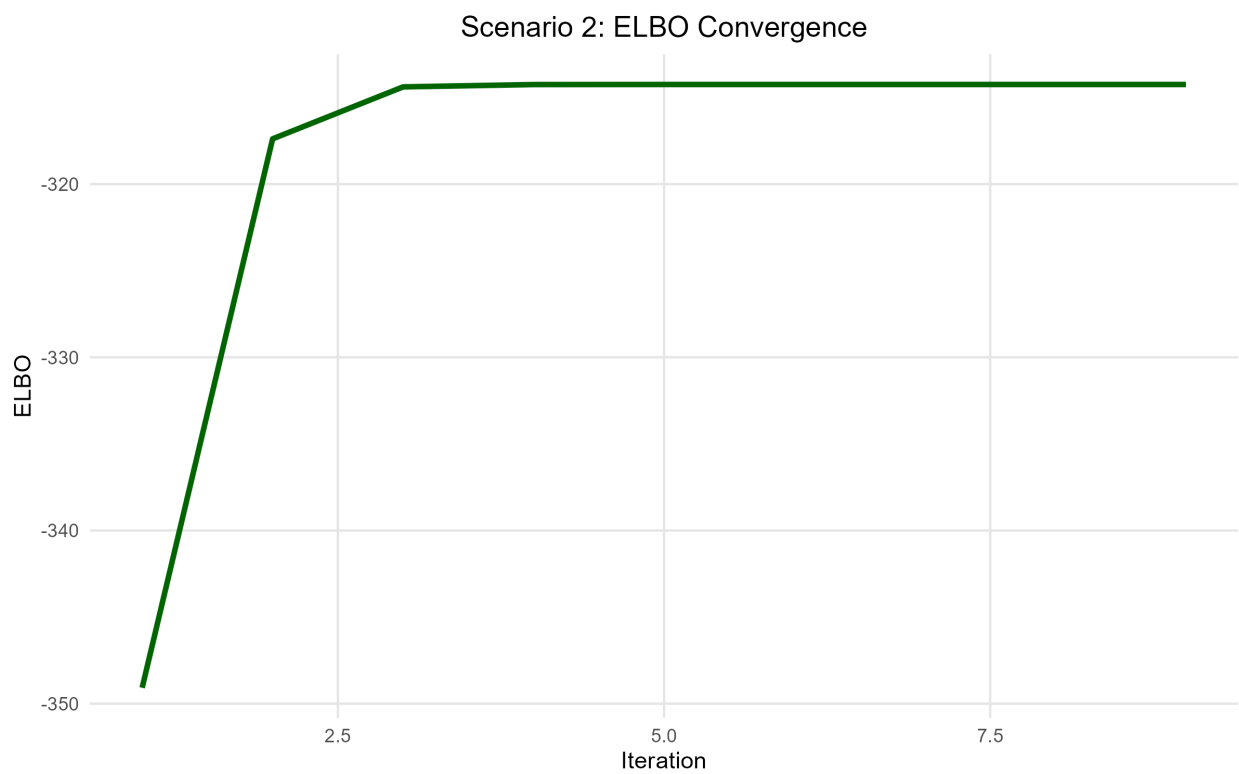
```

```

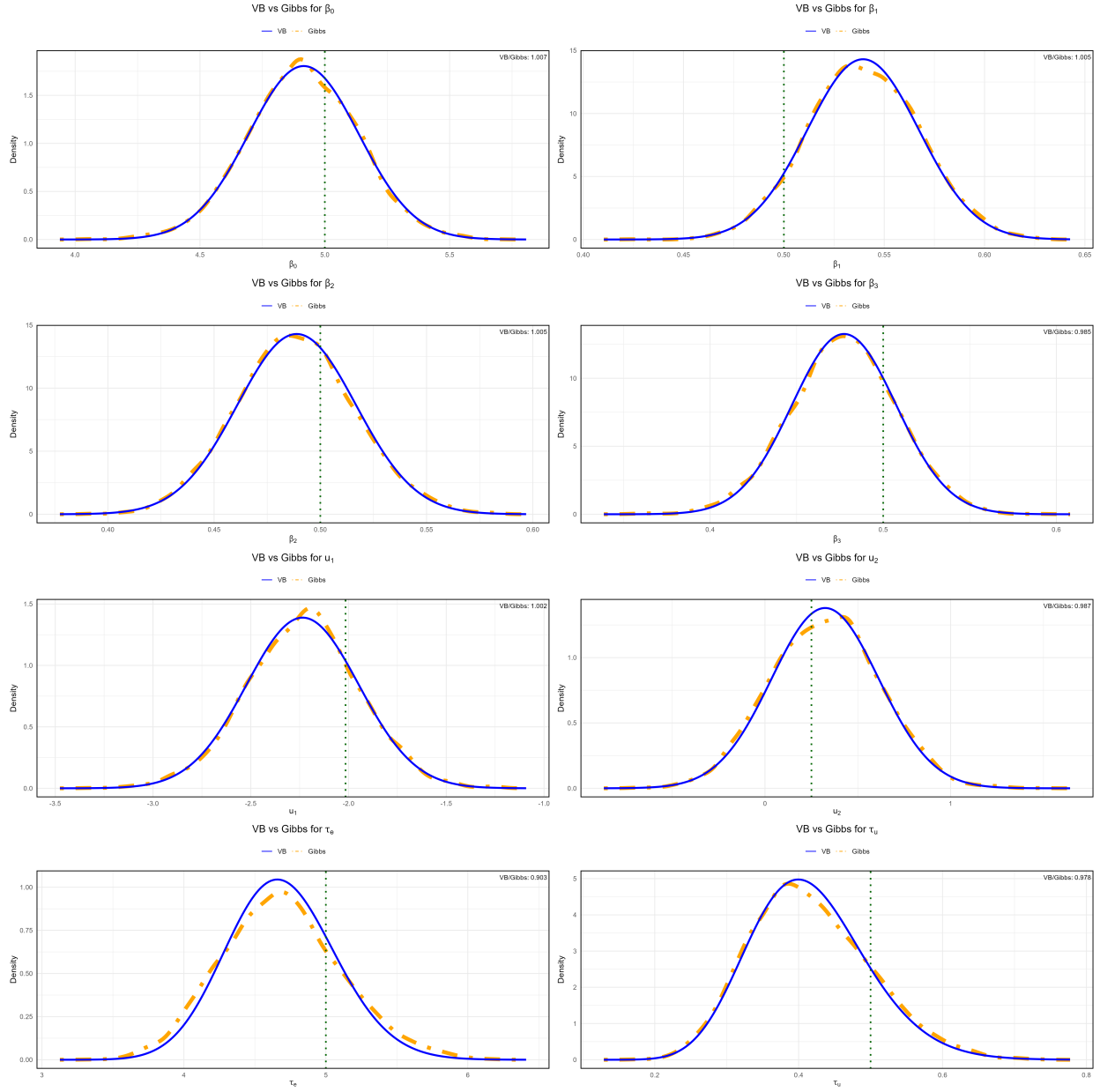
##
## Running Gibbs sampler...
## Gibbs posterior means:
## beta: 4.911552 0.5397689 0.4884797 0.4771542
## tau_e: 4.686553
## tau_u: 0.4160981

```





M3: 8-panel ggplot saved for Scenario 2



3 Comparison Between Scenarios (M3 only)

```
comparison_df <- data.frame(
  Parameter = c("E[tau_e]", "E[tau_u]", "sigma^2_e", "sigma^2_u"),
  True_Value = c(tau_e_true, tau_u_true, 1/tau_e_true, 1/tau_u_true),
  Scenario_30 = c(results_s1$E_tau_e, results_s1$E_tau_u,
                  1/results_s1$E_tau_e, 1/results_s1$E_tau_u),
  Scenario_6 = c(results_s2$E_tau_e, results_s2$E_tau_u,
                  1/results_s2$E_tau_e, 1/results_s2$E_tau_u)
)

print(comparison_df)
```

```
## Parameter True_Value Scenario_30 Scenario_6
## 1 E[tau_e]      5.0    5.0600907  4.6895713
## 2 E[tau_u]      0.5    0.8808896  0.4155009
## 3 sigma^2_e     0.2    0.1976249  0.2132391
## 4 sigma^2_u     2.0    1.1352161  2.4067335
```

```
cat("\nUnder-dispersion in tau_u estimates:\n")
```

```
##
## Under-dispersion in tau_u estimates:
```

```
cat("Q=5: VB tau_u =", round(results_s1$E_tau_u, 4),
    "vs True =", tau_u_true,
    "(ratio:", round(results_s1$E_tau_u / tau_u_true, 4), ")\n")
```

```
## Q=5: VB tau_u = 0.8809 vs True = 0.5 (ratio: 1.7618 )
```

```
cat("Q=50: VB tau_u =", round(results_s2$E_tau_u, 4),
    "vs True =", tau_u_true,
    "(ratio:", round(results_s2$E_tau_u / tau_u_true, 4), ")\n")
```

```
## Q=50: VB tau_u = 0.4155 vs True = 0.5 (ratio: 0.831 )
```

4 Sample Size Effects on τ_u (Multi-Configuration Analysis)

```
# Experimental design: Fix N=300, vary Q to show sample size per group effect
# Following Dr John's guidance (Meeting 16 Jan 2026)
# Q = 5 → 60 obs/group (rich data)
# Q = 10 → 30 obs/group
# Q = 20 → 15 obs/group
# Q = 50 → 6 obs/group (sparse data)
```

```
cat("\n===== \n")
```

```
##
## =====
```

```
cat("Multi-Configuration  $\tau_u$  Analysis\n")
```

```
## Multi-Configuration  $\tau_u$  Analysis
```

```
cat("===== \n")
```

```
## =====
```

```

group_configs <- list(
  list(q = 5, nq = 60, label = "Q=5 (n=60 per group)"),
  list(q = 10, nq = 30, label = "Q=10 (n=30 per group)"),
  list(q = 20, nq = 15, label = "Q=20 (n=15 per group)"),
  list(q = 50, nq = 6, label = "Q=50 (n=6 per group)")
)

results_multi <- list()

for (i in seq_along(group_configs)) {
  config <- group_configs[[i]]
  cat("\n--- Running:", config$label, "---\n")

  # Generate data
  q_temp <- config$q
  nq_temp <- config$nq

  set.seed(82171165 + i)

  # Design matrix for random effects
  Z_temp <- model.matrix(~ 0 + factor(rep(1:q_temp, each = nq_temp)))
  u_true_temp <- rnorm(q_temp, 0, sqrt(1/tau_u_true))

  # Correlation matrix for X
  Sigma_X <- matrix(c(
    1.0, 0.3, 0.2,
    0.3, 1.0, 0.4,
    0.2, 0.4, 1.0
  ), nrow = 3)

  X_raw <- mvtnorm::rmvnorm(n, mean = rep(0, 3), sigma = Sigma_X)
  X_temp <- cbind(1, X_raw)

  # Generate response
  eta_temp <- X_temp %*% beta_true + Z_temp %*% u_true_temp
  mu_temp <- 1 / (1 + exp(-eta_temp))
  y_temp <- rbinom(n, size = 1, prob = mu_temp)

  # Covariance matrix for random effects
  K_temp <- diag(q_temp)

  # Run VB
  vb_result <- run_vb_algorithm(
    X      = X_temp,
    Z      = Z_temp,
    y      = y_temp,
    K      = K_temp,
    p      = p,
    q      = q_temp,
    n      = n,
    alpha_e = alpha_e,
    gamma_e = gamma_e,
    alpha_u = alpha_u,

```

```

    gamma_u = gamma_u,
    model_type = "M3",
    tol      = 1e-4,
    max_iter = 500
  )

  # Run HMC if enabled
  if (run_gibbs) {
    gibbs_result <- run_gibbs_sampler(
      X      = X_temp,
      Z      = Z_temp,
      y      = y_temp,
      p      = p,
      q      = q_temp,
      n      = n,
      alpha_e = alpha_e,
      gamma_e = gamma_e,
      alpha_u = alpha_u,
      gamma_u = gamma_u,
      model_type = "M3",
      n_iter    = gibbs_iter,
      n_burnin  = gibbs_burnin
    )
  } else {
    gibbs_result <- NULL
  }

  # Store results
  results_multi[[i]] <- list(
    config = config,
    vb     = vb_result,
    gibbs  = gibbs_result
  )

  cat("VB E[tau_u]:", round(vb_result$E_tau_u, 4), "\n")
  if (!is.null(gibbs_result)) {
    cat("HMC E[tau_u]:", round(mean(gibbs_result[, "tau_u"]), 4), "\n")
  }
}

##
## --- Running: Q=5 (n=60 per group) ---
## Converged at iteration 8
## Max relative change: 1.44e-05
## VB E[tau_u]: 1.4992
## HMC E[tau_u]: 1.5192
##
## --- Running: Q=10 (n=30 per group) ---
## Converged at iteration 7
## Max relative change: 1.71e-05
## VB E[tau_u]: 2.6993
## HMC E[tau_u]: 2.6879
##
## --- Running: Q=20 (n=15 per group) ---

```

```

## Converged at iteration 7
## Max relative change: 1.59e-05
## VB E[tau_u]: 5.0595
## HMC E[tau_u]: 5.0934
##
## --- Running: Q=50 (n=6 per group) ---
## Converged at iteration 9
## Max relative change: 2.64e-05
## VB E[tau_u]: 11.2881
## HMC E[tau_u]: 11.2815

cat("\n===== \n")

##
## =====

par(mfrow = c(1, 2))

# E[tau_u] convergence for all 4 configurations
max_len_tau <- max(sapply(results_multi, function(r) length(r$vb$E_tau_u_history)))

plot(
  1:length(results_multi[[1]]$vb$E_tau_u_history),
  results_multi[[1]]$vb$E_tau_u_history,
  type = 'l',
  lwd = 2,
  col = '#1b9e77',
  xlab = 'Iteration',
  ylab = 'E[tau_u]',
  main = 'Comparison: E[tau_u] Convergence (varying Q, fixed N=300)',
  xlim = c(1, max_len_tau),
  ylim = range(c(sapply(results_multi, function(r) r$vb$E_tau_u_history), tau_u_true)))

lines(1:length(results_multi[[2]]$vb$E_tau_u_history),
      results_multi[[2]]$vb$E_tau_u_history,
      col = '#d95f02', lwd = 2)
lines(1:length(results_multi[[3]]$vb$E_tau_u_history),
      results_multi[[3]]$vb$E_tau_u_history,
      col = '#7570b3', lwd = 2)
lines(1:length(results_multi[[4]]$vb$E_tau_u_history),
      results_multi[[4]]$vb$E_tau_u_history,
      col = '#e7298a', lwd = 2)

abline(h = tau_u_true, col = 'red', lty = 2, lwd = 2)

legend('topright',
      legend = c('Q=5 (n=60)', 'Q=10 (n=30)', 'Q=20 (n=15)', 'Q=50 (n=6)', 'True value'),
      col = c('#1b9e77', '#d95f02', '#7570b3', '#e7298a', 'red'),
      lty = c(1, 1, 1, 1, 2),
      lwd = 2,
      cex = 0.8)

# ELBO convergence for all 4 configurations
max_len_elbo <- max(sapply(results_multi, function(r) length(r$vb$elbo_history)))

```

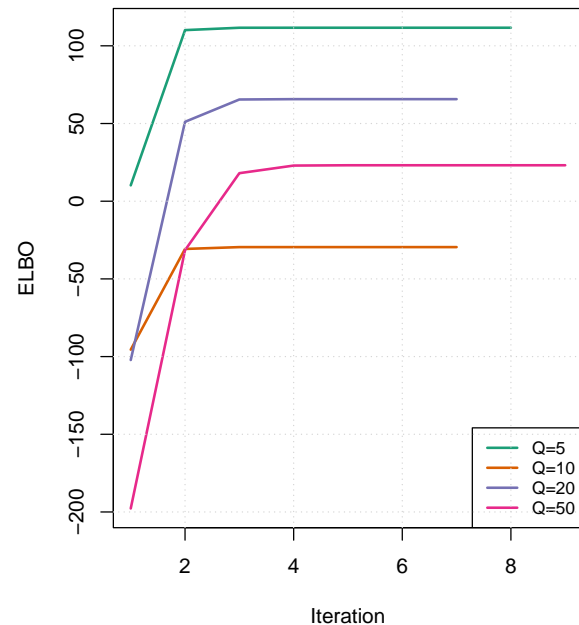
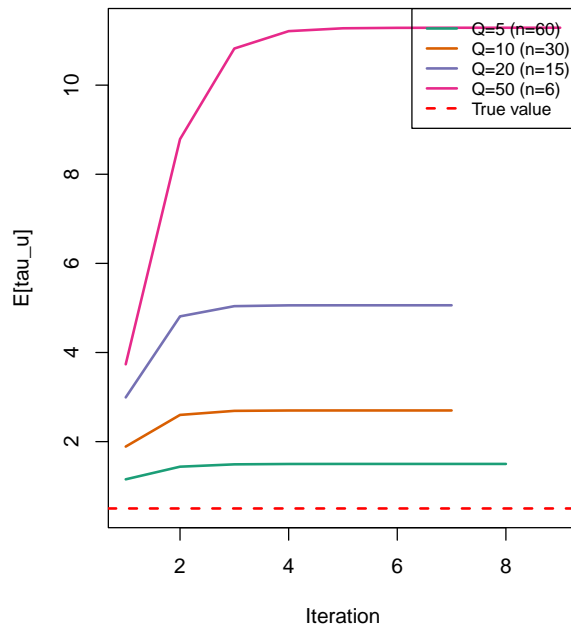
```
plot(
  1:length(results_multi[[1]]$vb$elbo_history),
  results_multi[[1]]$vb$elbo_history,
  type = 'l',
  lwd = 2,
  col = '#1b9e77',
  xlab = 'Iteration',
  ylab = 'ELBO',
  main = 'Comparison: ELBO Convergence (varying Q, fixed N=300)',
  xlim = c(1, max_len_elbo),
  ylim = range(sapply(results_multi, function(r) r$vb$elbo_history)))

lines(1:length(results_multi[[2]]$vb$elbo_history),
      results_multi[[2]]$vb$elbo_history,
      col = '#d95f02', lwd = 2)
lines(1:length(results_multi[[3]]$vb$elbo_history),
      results_multi[[3]]$vb$elbo_history,
      col = '#7570b3', lwd = 2)
lines(1:length(results_multi[[4]]$vb$elbo_history),
      results_multi[[4]]$vb$elbo_history,
      col = '#e7298a', lwd = 2)

legend('bottomright',
      legend = c('Q=5', 'Q=10', 'Q=20', 'Q=50'),
      col = c('#1b9e77', '#d95f02', '#7570b3', '#e7298a'),
      lty = 1,
      lwd = 2,
      cex = 0.8)

grid()
```

Comparison: E[tau_u] Convergence (varying Q, fixed N=300) Comparison: ELBO Convergence (varying Q, fixed N=300)



```

# Create multi-panel comparison plot as requested by Dr John
# Focus on _u posterior distributions across different group sizes

plot_list <- list()

for (i in seq_along(results_multi)) {
  result <- results_multi[[i]]
  config <- result$config

  # VB posterior: Gamma(a_u_new, b_u_new)
  a_vb <- result$vb$a_u_new
  b_vb <- result$vb$b_u_new

  # Calculate VB range
  vb_mean <- a_vb / b_vb
  vb_sd <- sqrt(a_vb) / b_vb
  vb_min <- max(0, vb_mean - 4 * vb_sd)
  vb_max <- vb_mean + 4 * vb_sd

  # If HMC available, extend range to include both distributions
  if (!is.null(result$gibbs)) {
    gibbs_tau_u <- result$gibbs[, "tau_u"]
    hmc_min <- quantile(gibbs_tau_u, 0.001)
    hmc_max <- quantile(gibbs_tau_u, 0.999)

    x_min <- min(vb_min, hmc_min, tau_u_true * 0.5)
    x_max <- max(vb_max, hmc_max, tau_u_true * 3)
  } else {
    x_min <- min(vb_min, tau_u_true * 0.5)
    x_max <- max(vb_max, tau_u_true * 3)
  }

  x_range <- seq(x_min, x_max, length.out = 500)
  vb_density <- dgamma(x_range, shape = a_vb, rate = b_vb)

  df_plot <- data.frame(
    tau_u = x_range,
    density = vb_density,
    method = "VB",
    type = "solid"
  )

  # Add Gibbs if available and calculate SD ratio
  sd_ratio_text <- ""
  if (!is.null(result$gibbs)) {
    dens_gibbs <- density(gibbs_tau_u, adjust = 1.5)

    df_gibbs <- data.frame(
      tau_u = dens_gibbs$x,
      density = dens_gibbs$y,
      method = "Gibbs",
      type = "dashed"
    )
  }
}

```



```

df_plot <- rbind(df_plot, df_gibbs)

# Calculate SD ratio
vb_sd <- sqrt(a_vb) / b_vb
gibbs_sd <- sd(gibbs_tau_u)
sd_ratio <- vb_sd / gibbs_sd
sd_ratio_text <- glue(" | SD ratio: {round(sd_ratio, 3)}")
}

# Create plot
p_tau <- ggplot(df_plot, aes(x = tau_u, y = density, color = method, linetype = method)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = tau_u_true, color = "red", linetype = "dotted", linewidth = 0.8) +
  scale_color_manual(
    values = c("VB" = "black", "Gibbs" = "#E7298A")
  ) +
  scale_linetype_manual(
    values = c("VB" = "solid", "Gibbs" = "dashed")
  ) +
  labs(
    title = config$label,
    subtitle = glue("VB E[_u] = {round(result$vb$E_tau_u, 3)}{sd_ratio_text}"),
    x = expression(tau[u]),
    y = "Density"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top",
    plot.title = element_text(size = 12, face = "bold"),
    plot.subtitle = element_text(size = 10)
  )

plot_list[[i]] <- p_tau
}

# Combine into 2x2 grid
combined_tau_u <- (plot_list[[1]] | plot_list[[2]]) /
  (plot_list[[3]] | plot_list[[4]]) +
  plot_annotation(
    title = "Effect of Sample Size Per Group on _u Posterior",
    subtitle = "VB approximation improves as observations per random effect level increase",
    theme = theme(
      plot.title = element_text(size = 16, face = "bold"),
      plot.subtitle = element_text(size = 12)
    )
  )

# Save plot
ggsave(
  filename = "../figs/tau_u_sample_size_comparison.png",
  plot = combined_tau_u,
  width = 14,
  height = 10,

```

```

    dpi      = 300
  )

  cat("_u comparison plot saved to figs/tau_u_sample_size_comparison.png\n")

## _u comparison plot saved to figs/tau_u_sample_size_comparison.png

# Display
img_tau_u <- readPNG("../figs/tau_u_sample_size_comparison.png")
grid.newpage()
grid.raster(img_tau_u)

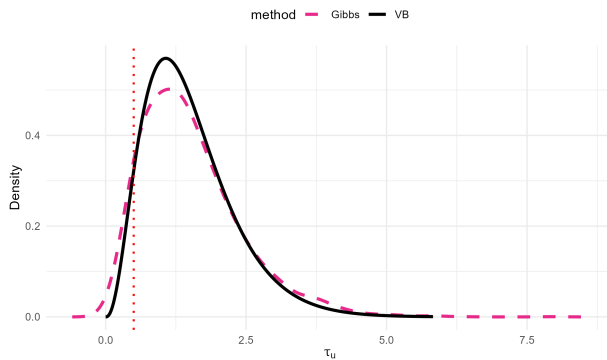
```

Effect of Sample Size Per Group on τ_u Posterior

VB approximation improves as observations per random effect level increase

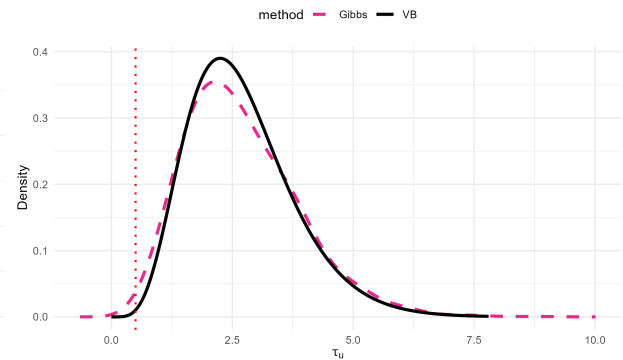
Q=5 (n=60 per group)

VB E[τ_u] = 1.499 | SD ratio: 0.902



Q=10 (n=30 per group)

VB E[τ_u] = 2.699 | SD ratio: 0.942



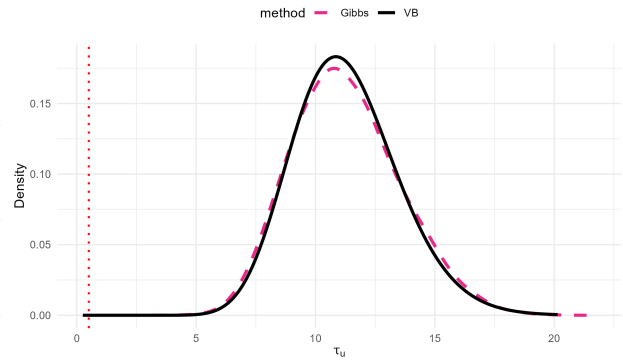
Q=20 (n=15 per group)

VB E[τ_u] = 5.059 | SD ratio: 0.979



Q=50 (n=6 per group)

VB E[τ_u] = 11.288 | SD ratio: 0.993



Create 2-panel overlay plot: All Gibbs together, All VB together

Prepare data for Gibbs panel

```

if (run_gibbs) {
  gibbs_combined <- data.frame()

  for (i in seq_along(results_multi)) {
    result <- results_multi[[i]]
    config <- result$config
    gibbs_tau_u <- result$gibbs[, "tau_u"]

    dens_gibbs <- density(gibbs_tau_u, adjust = 1.5)

    df_temp <- data.frame(

```

```

    tau_u    = dens_gibbs$x,
    density  = dens_gibbs$y,
    config   = config$label
  )

  gibbs_combined <- rbind(gibbs_combined, df_temp)
}

# Gibbs panel
p_gibbs <- ggplot(gibbs_combined, aes(x = tau_u, y = density, color = config)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = tau_u_true, color = "red", linetype = "dotted", linewidth = 0.8) +
  scale_color_manual(
    values = c(
      "Q=5 (n=60 per group)" = "#1b9e77",
      "Q=10 (n=30 per group)" = "#d95f02",
      "Q=20 (n=15 per group)" = "#7570b3",
      "Q=50 (n=6 per group)" = "#e7298a"
    ),
    guide = guide_legend(nrow = 1, title.position = "left")
  ) +
  labs(
    title = "Gibbs Sampling Posteriors",
    subtitle = "All configurations show similar distributions",
    x = expression(tau[u]),
    y = "Density",
    color = "Configuration"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top",
    legend.text = element_text(size = 9),
    legend.title = element_text(size = 10, face = "bold"),
    legend.key.width = unit(1.5, "cm"),
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 11)
  )
}

# Prepare data for VB panel
vb_combined <- data.frame()

for (i in seq_along(results_multi)) {
  result <- results_multi[[i]]
  config <- result$config

  a_vb <- result$vb$a_u_new
  b_vb <- result$vb$b_u_new

  # Use broad range to show all VB distributions
  x_range <- seq(0, 20, length.out = 500)
  vb_density <- dgamma(x_range, shape = a_vb, rate = b_vb)

```

```

df_temp <- data.frame(
  tau_u    = x_range,
  density  = vb_density,
  config   = config$label
)

vb_combined <- rbind(vb_combined, df_temp)
}

# VB panel
p_vb <- ggplot(vb_combined, aes(x = tau_u, y = density, color = config)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = tau_u_true, color = "red", linetype = "dotted", linewidth = 0.8) +
  scale_color_manual(
    values = c(
      "Q=5 (n=60 per group)" = "#1b9e77",
      "Q=10 (n=30 per group)" = "#d95f02",
      "Q=20 (n=15 per group)" = "#7570b3",
      "Q=50 (n=6 per group)" = "#e7298a"
    ),
    guide = guide_legend(nrow = 1, title.position = "left")
) +
  labs(
    title = "VB Posteriors",
    subtitle = "Large variation across configurations",
    x = expression(tau[u]),
    y = "Density",
    color = "Configuration"
) +
  theme_minimal() +
  theme(
    legend.position = "top",
    legend.text = element_text(size = 9),
    legend.title = element_text(size = 10, face = "bold"),
    legend.key.width = unit(1.5, "cm"),
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 11)
)

# Combine panels
if (run_gibbs) {
  combined_overlay <- p_gibbs | p_vb
  plot_title <- "Comparison: Gibbs vs VB Across All Configurations"
  plot_subtitle <- "Gibbs posteriors are consistent; VB posteriors vary dramatically with sample size per group"
  plot_width <- 14
} else {
  combined_overlay <- p_vb
  plot_title <- "VB Posteriors Across All Configurations"
  plot_subtitle <- "VB posterior quality varies dramatically with sample size per group"
  plot_width <- 8
}

combined_overlay <- combined_overlay +

```

```

plot_annotation(
  title = plot_title,
  subtitle = plot_subtitle,
  theme = theme(
    plot.title = element_text(size = 16, face = "bold", margin = margin(b = 10)),
    plot.subtitle = element_text(size = 12, margin = margin(b = 20))
  )
) &
theme(
  legend.position = "top",
  legend.box = "horizontal",
  legend.margin = margin(t = 10, b = 15),
  legend.spacing.x = unit(0.5, "cm"),
  plot.margin = margin(t = 15, r = 10, b = 10, l = 10)
)

# Save plot
ggsave(
  filename = "../figs/tau_u_overlay_comparison.png",
  plot = combined_overlay,
  width = plot_width,
  height = 7,
  dpi = 300
)

cat("_u overlay comparison plot saved to figs/tau_u_overlay_comparison.png\n")

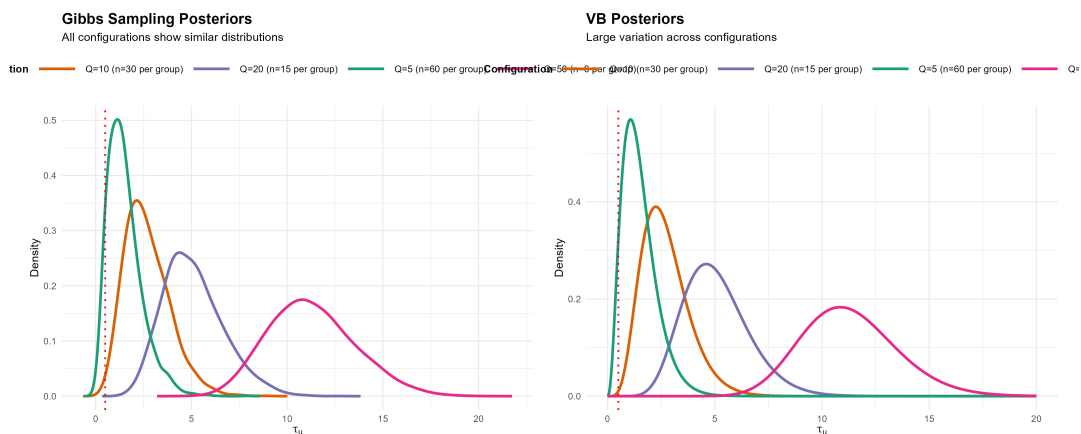
## _u overlay comparison plot saved to figs/tau_u_overlay_comparison.png

# Display
img_overlay <- readPNG("../figs/tau_u_overlay_comparison.png")
grid.newpage()
grid.raster(img_overlay)

```

Comparison: Gibbs vs VB Across All Configurations

Gibbs posteriors are consistent; VB posteriors vary dramatically with sample size per group



```

# Diagnostic: Posterior variance / Prior variance ratio for u's
# As requested by Dr John [0:15:49]
# "When you do badly with tau_u, this ratio will be high. When you do well, this ratio will be low."

```

```

cat("\n===== \n")

##
## =====

cat("Diagnostic: Var_posterior(u) / Var_prior(u)\n")

## Diagnostic: Var_posterior(u) / Var_prior(u)
cat("===== \n\n")

## =====

# Prior variance:  $u_i \sim N(0, 1/\tau_{u\_true})$ 
var_prior_u <- 1 / tau_u_true

# Calculate ratio for each configuration
ratio_data <- data.frame()

for (i in seq_along(results_multi)) {
  result <- results_multi[[i]]
  config <- result$config
  q <- config$q

  # VB posterior variances: diagonal of Sigma_betau for u's
  Sigma_betau <- result$vb$Sigma_betau
  var_post_vb_u <- diag(Sigma_betau)[(p+1):(p+q)]
  mean_ratio_vb <- mean(var_post_vb_u / var_prior_u)

  # Gibbs posterior variances if available
  if (!is.null(result$gibbs)) {
    var_post_gibbs_u <- sapply(1:q, function(j) {
      var(result$gibbs[, paste0("u", j)])
    })
    mean_ratio_gibbs <- mean(var_post_gibbs_u / var_prior_u)
  } else {
    mean_ratio_gibbs <- NA
  }

  # Store results
  ratio_data <- rbind(ratio_data, data.frame(
    Q = q,
    n_per_group = config$nq,
    VB_ratio = mean_ratio_vb,
    Gibbs_ratio = mean_ratio_gibbs,
    label = config$label
  ))
}

print(ratio_data)

##      Q n_per_group  VB_ratio Gibbs_ratio      label
## 1   5             60 0.066852623 0.096521749 Q=5 (n=60 per group)
## 2  10             30 0.019339286 0.023605108 Q=10 (n=30 per group)
## 3  20             15 0.005727350 0.006054534 Q=20 (n=15 per group)
## 4  50              6 0.003035513 0.003069926 Q=50 (n=6 per group)

```

```

cat("\nInterpretation:\n")

##
## Interpretation:
cat("- Lower ratio = narrower posteriors = more information learnt\n")

## - Lower ratio = narrower posteriors = more information learnt
cat("- Narrow posteriors for u → better tau_u estimation in VB\n")

## - Narrow posteriors for u → better tau_u estimation in VB
cat("- As n_per_group increases, VB ratio decreases (posteriors concentrate)\n\n")

## - As n_per_group increases, VB ratio decreases (posteriors concentrate)

# Prepare data for plotting
plot_data <- data.frame(
  Q = ratio_data$Q,
  VB = ratio_data$VB_ratio
)

if (run_gibbs) {
  plot_data$Gibbs_ratio <- ratio_data$Gibbs_ratio
  plot_data_long <- tidyr::pivot_longer(plot_data, cols = c(VB, Gibbs),
                                         names_to = "Method", values_to = "Ratio")
} else {
  plot_data_long <- data.frame(
    Q = plot_data$Q,
    Method = "VB",
    Ratio = plot_data$VB
  )
}

# Create diagnostic plot
p_diagnostic <- ggplot(plot_data_long, aes(x = factor(Q), y = Ratio, color = Method, group = Method)) +
  geom_point(size = 4) +
  geom_line(aes(linetype = Method), size = 1.2) +
  scale_color_manual(
    values = c("VB" = "black", "Gibbs" = "#E7298A")
  ) +
  scale_linetype_manual(
    values = c("VB" = "solid", "Gibbs" = "dashed")
  ) +
  labs(
    title = "Diagnostic: Posterior Variance / Prior Variance Ratio for Random Effects",
    subtitle = "Varying Q (fixed N=300): Lower ratio indicates concentrated posteriors and better _u es",
    x = "Number of Groups (Q) [n per group = 300/Q]",
    y = "Mean(Var_posterior(u) / Var_prior(u))",
    color = "Method"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top",
    plot.title = element_text(size = 14, face = "bold"),

```

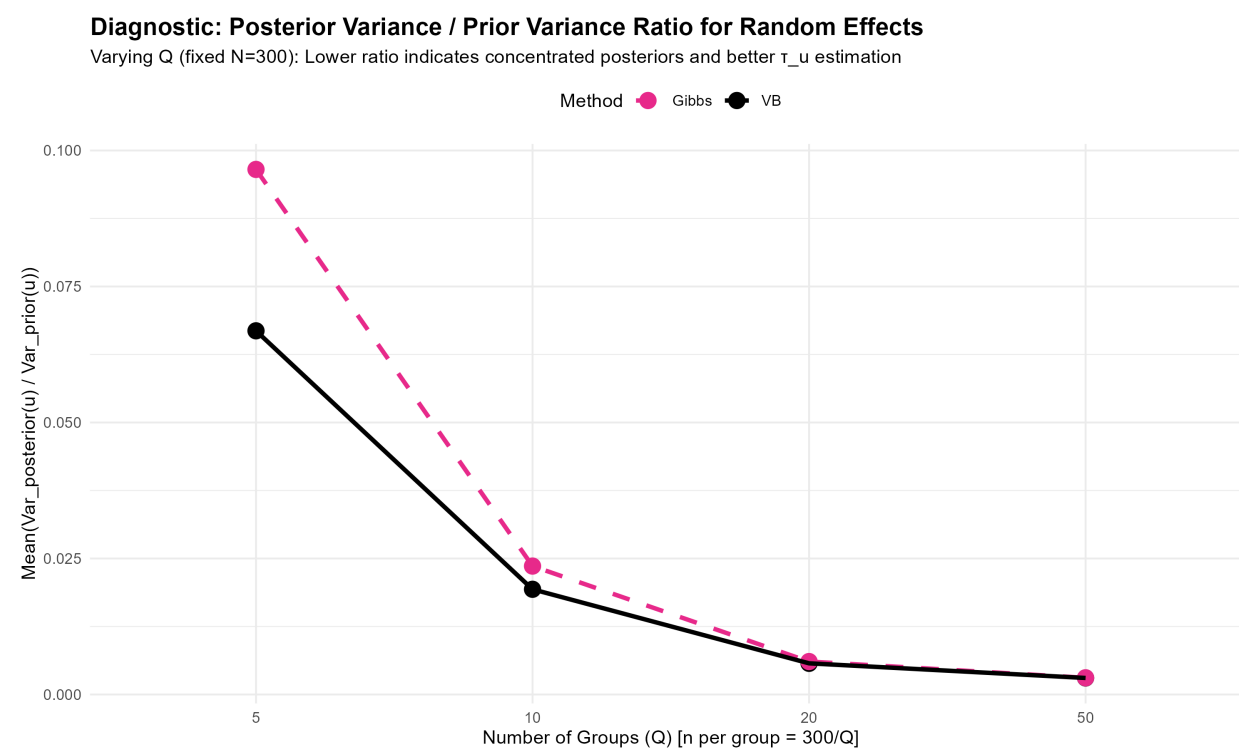
```

    plot.subtitle = element_text(size = 11)
  )

  ggsave(
    filename = "../figs/diagnostic_variance_ratio.png",
    plot = p_diagnostic,
    width = 10,
    height = 6,
    dpi = 300
  )

  img_diagnostic <- readPNG("../figs/diagnostic_variance_ratio.png")
  grid.newpage()
  grid.raster(img_diagnostic)

```



```

cat("\n===== \n")

##
## =====

cat("Key Finding (Dr John's insight):\n")

## Key Finding (Dr John's insight):
cat("As n per group increases (Q decreases from 50→5),\n")

## As n per group increases (Q decreases from 50→5),
cat("VB posteriors for u become narrower (ratio decreases),\n")

## VB posteriors for u become narrower (ratio decreases),

```



```
cat("leading to better tau_u estimation.\n")

## leading to better tau_u estimation.
cat("=====\n")

## =====
```