

Model 0: Hierarchical Linear Reference Implementation

M0-Hierarchical.Rmd | Compiled: 2026-01-26 09:49:12

Contents

1	Multi-Configuration Analysis: μ Under-Dispersion Across Q	6
1.1	Experimental Design	6
2	Aggregate Analysis Across Configurations	12
2.1	Three-Panel Validation Comparison	25

M0-Hierarchical.Rmd | Compiled: 2026-01-26 09:49:12

```
# =====
# MODEL 0: HIERARCHICAL LINEAR REFERENCE IMPLEMENTATION
# Dr John Holmes' original VB and Gibbs code - no diagnostic modifications
# Hierarchical model:  $y = X + Zu +$ 
# Parameters: (fixed effects), u (random effects), (residual precision), (variance component)
# Demonstrates variance component under-dispersion across multiple Q configurations
# =====

# MODEL IDENTIFIER (prevents plot filename collisions between M0/M1/M2/M3)
model_prefix <- "M0_"

# FIXED MODEL PARAMETERS
n <- 300 # Total observations
p <- 3   # Number of fixed effects ( , , ) - Dr John's setup

# TRUE PARAMETER VALUES (Dr John's exact values)
tau_e_true <- 0.5 # Residual precision (variance = 2)
tau_u_true <- 1   # Random effect precision (variance = 1)
beta_true  <- c(0.5, -2, 3) # Dr John's values

# prior Hyperparameters (Dr John's flat priors - no prior information)
alpha_e <- 0
gamma_e <- 0
alpha_u <- 0
gamma_u <- 0

# 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 CONFIGURATIONS (varying group structure)
# SCENARIO 1: Q=5 groups (n=60 per group)
q_s1 <- 5
nq_s1 <- 60

# SCENARIO 2: Q=10 groups (n=30 per group)
q_s2 <- 10
nq_s2 <- 30

# SCENARIO 3: Q=20 groups (n=15 per group)
q_s3 <- 20
nq_s3 <- 15

# SCENARIO 4: Q=50 groups (n=6 per group)
q_s4 <- 50
nq_s4 <- 6

# SCENARIO 5: Q=100 groups (n=3 per group)
q_s5 <- 100
```

```
nq_s5 <- 3
```

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

```

# Dr John Holmes' Gibbs sampler for hierarchical models
# EXACT original from Dr John - NO modifications
normalmm.Gibbs<-function(iter,Z,X,y,burnin,taue_0,tauu_0,Kinv,a.u,b.u,a.e,b.e){
  n <-length(y) #no. observations
  p <-dim(X)[2] #no of fixed effect predictors.
  q <-dim(Z)[2] #no of random effect levels
  tauu<-tauu_0
  taue<-taue_0
  beta0<-rnorm(p)
  u0 <-rnorm(q,0,sd=1/sqrt(tauu))

  #Building combined predictor matrix.
  W<-cbind(X,Z)
  WTW <-crossprod(W)
  WTy <-crossprod(W,y)
  library(mvtnorm)

  #storing results.
  par <-matrix(0,iter,p+q+2)
  #Calculating log predictive densities
  lppd<-matrix(0,iter,n)

  #Create modified identity matrix for joint posterior.
  IO <-diag(p+q)
  diag(IO)[1:p]<-0
  IO[-c(1:p),-c(1:p)] <-Kinv

  for(i in 1:iter){
    #Conditional posteriors.
    uKinvu <- t(u0)%*%Kinv%*%u0
    uKinvu <-as.numeric(uKinvu)
    tauu <-rgamma(1,a.u+0.5*q,b.u+0.5*uKinvu)
    #Updating component of normal posterior for beta,u
    Prec <-WTW*taue + tauu*IO
    P.var <-solve(Prec)
    P.mean<- P.var%*%WTy*taue
    betau <-rmvnorm(1,mean=P.mean,sigma=P.var)
    betau <-as.numeric(betau)
    err <- y-W%*%betau
    taue <-rgamma(1,a.e+0.5*n,b.e+0.5*sum(err^2))
    #storing iterations.
    par[i,]<-c(betau,tauu,taue)
    beta0 <-betau[1:p]
    u0 <-betau[p+1:q]
    lppd[i,]= dnorm(y,mean=as.numeric(W%*%betau),sd=1/sqrt(taue))
  }

  lppd = lppd[-c(1:burnin),]
  lppdest = sum(log(colMeans(lppd))) #Estimating lppd for whole dataset.
  pwaic2 = sum(apply(log(lppd),2,FUN=var)) #Estimating effective number of parameters.
  par <-par[-c(1:burnin),]
  colnames(par)<-c(paste('beta',1:p,sep=' '),paste('u',1:q,sep=' '), 'tau_u', 'tau_e')
  mresult<-list(par,lppdest,pwaic2)
}

```

```

names(mresult)<-c('par','lppd','pwaic')
return(mresult)
}

# Dr John Holmes' VB algorithm for hierarchical models
# EXACT original from Dr John - NO modifications
VB.mm<-function(epsilon,iter,Kinv,Z,X,y,taue_0,tauu_0,u0,beta0,a.e,g.e,a.u,g.u){
  n<-dim(X)[1]
  p<-dim(X)[2]
  q<-dim(Z)[2]
  W <-cbind(X,Z)
  WTW<-crossprod(W)
  WTY<-crossprod(W,y)
  Kinvall<-matrix(0,p+q,p+q)
  Kinvall[-c(1:p),-c(1:p)]<-Kinv

  for(i in 1:iter){
    Vub <-solve(taue_0*WTW+tauu_0*Kinvall) #update Var(b,u)
    ub <-taue_0*Vub%%WTY #update E(b,u)
    TrKinub <- sum(diag(Kinvall%%Vub))
    uKinub <- t(ub)%%Kinvall%%ub
    tauu <- (a.u+0.5*q)/(g.u+0.5*as.numeric(uKinub)+0.5*TrKinub)
    tauu <- as.numeric(tauu)
    err <- y - W%%ub
    TrWTWub <- sum(diag(WTW%%Vub))
    taue <- (a.e+0.5*n)/(g.e+0.5*sum(err^2)+0.5*TrWTWub)
    taue <- as.numeric(taue)

    if(i > 1){
      diffub <- sqrt((ub-ub0)^2)/(abs(ub)+0.01)
      diffte <- abs(taue_0-taue)/(taue+0.01)
      difftu <- abs(tauu_0-tauu)/(tauu+0.01)
      diffvub <- sqrt((diag(Vub0) - diag(Vub))^2)/(diag(Vub))
      diff.all<-c(diffub,diffte,diftu,diffvub)
      if(max(diff.all) < epsilon) break
    }
    Vub0 <- Vub;ub0<-ub;taue_0<-taue;tauu_0<-tauu
    #Calculate relative change.
  }

  taue.param<-c((a.e+0.5*n),(g.e+0.5*sum(err^2)+0.5*TrWTWub))
  tauu.param<-c((a.u+0.5*q),(g.u+0.5*uKinub+0.5*TrKinub))
  param<-list(ub,Vub,taue.param,tauu.param,i)
  names(param)<-c('betau_mean','betau_var','tau_e','tau_u','iter')
  return(param)
}

```

1 Multi-Configuration Analysis: $_u$ Under-Dispersion Across Q

1.1 Experimental Design

Fix $N=300$, vary Q to show sample size per group effect. Following Dr John's guidance (Meeting 16 Jan 2026):

- $Q = 5 \rightarrow 60$ obs/group (rich data)
- $Q = 10 \rightarrow 30$ obs/group
- $Q = 20 \rightarrow 15$ obs/group
- $Q = 50 \rightarrow 6$ obs/group
- $Q = 100 \rightarrow 3$ obs/group (sparse data)

```
# 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  $\_u$  Analysis\n")
```

```
## Multi-Configuration  $\_u$  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)"),
  list(q = 100, nq = 3, label = "Q=100 (n=3 per group)")
)
```

```
results_multi <- list()
```

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

```
  # generate data using Dr John's exact method
```

```
  q_temp <- config$q
```

```
  nq_temp <- config$nq
```

```
  # Dr John's method: generate then standardize
```

```
  u_true_temp <- rnorm(q_temp, 0, 1)
```

```
  u_true_temp <- scale(u_true_temp)
```

```
  # Dr John's method: table() for incidence matrix
```

```
  Z_temp <- table(1:n, rep(1:q_temp, n/q_temp))
```

```

# Dr John's method: simple X matrix (no correlation)
X_temp <- cbind(1, matrix(rnorm(n*(p-1)), n, p-1))

# Dr John's method: hardcoded residual sd = sqrt(2)
eta_temp <- X_temp %*% beta_true + Z_temp %*% u_true_temp
y_temp <- as.vector(eta_temp + rnorm(n, 0, sqrt(2)))

# 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,

  tol      = 1e-4,
  max_iter = 500
)

# Run Gibbs (Dr John's sampler already runs 3 chains with different inits internally)
if (run_gibbs) {
  cat("Running Gibbs with 3 different initial values...\n")
  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,

    n_iter  = gibbs_iter,
    n_burnin = gibbs_burnin
  )
  cat("Combined", nrow(gibbs_result), "samples from 3 Gibbs chains\n")
} 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")
  }

  # Generate convergence plot for this configuration
  scenario_name <- glue("Configuration {i}: Q={q_temp} (n={nq_temp} per group)")
  scenario_label <- glue("s{i}")

  p_conv <- plot_convergence(
    results      = vb_result,
    scenario_name = scenario_name,
    tau_e_true   = tau_e_true,
    tau_u_true   = tau_u_true
  )
  ggsave(glue("../figs/{model_prefix}{scenario_label}_convergence.png"), plot = p_conv, width = 10, height = 10)

  # Generate ELBO plot for this configuration
  p_elbo <- plot_elbo(
    results      = vb_result,
    scenario_name = scenario_name
  )
  ggsave(glue("../figs/{model_prefix}{scenario_label}_elbo.png"), plot = p_elbo, width = 6, height = 4, units = "in")

  # Generate 8-panel posterior plot for this configuration
  Sigma_betau <- vb_result$Sigma_betau
  mu_betau    <- vb_result$mu_betau

  # VB samples
  beta_samples_vb <- rmvnorm(5000, mean = mu_betau[1:p], sigma = Sigma_betau[1:p, 1:p])
  u_samples_vb    <- rmvnorm(5000, mean = mu_betau[(p+1):(p+q_temp)], sigma = Sigma_betau[(p+1):(p+q_temp), (p+1):(p+q_temp)])
  tau_e_samples_vb <- rgamma(5000, shape = vb_result$a_e_new, rate = vb_result$b_e_new)
  tau_u_samples_vb <- rgamma(5000, shape = vb_result$a_u_new, rate = vb_result$b_u_new)

  # Data frames for VB
  df_beta0_vb <- data.frame(value = beta_samples_vb[, 1], method = "VB", param = "beta0")
  df_beta1_vb <- data.frame(value = beta_samples_vb[, 2], method = "VB", param = "beta1")
  df_tau_e_vb  <- data.frame(value = tau_e_samples_vb, method = "VB", param = "tau_e")
  df_tau_u_vb  <- data.frame(value = tau_u_samples_vb, method = "VB", param = "tau_u")

  # Add Gibbs if available
  if (!is.null(gibbs_result)) {
    df_beta0_gibbs <- data.frame(value = gibbs_result[, "beta0"], method = "HMC", param = "beta0")
    df_beta1_gibbs <- data.frame(value = gibbs_result[, "beta1"], method = "HMC", param = "beta1")
    df_tau_e_gibbs  <- data.frame(value = gibbs_result[, "tau_e"], method = "HMC", param = "tau_e")
    df_tau_u_gibbs  <- data.frame(value = gibbs_result[, "tau_u"], method = "HMC", param = "tau_u")

    df_beta0_combined <- rbind(df_beta0_vb, df_beta0_gibbs)
  }

```



```

df_beta1_combined <- rbind(df_beta1_vb, df_beta1_gibbs)
df_tau_e_combined <- rbind(df_tau_e_vb, df_tau_e_gibbs)
df_tau_u_combined <- rbind(df_tau_u_vb, df_tau_u_gibbs)
} else {
  df_beta0_combined <- df_beta0_vb
  df_beta1_combined <- df_beta1_vb
  df_tau_e_combined <- df_tau_e_vb
  df_tau_u_combined <- df_tau_u_vb
}

# Create 8 panels
p1 <- ggplot(df_beta0_vb, aes(x = value)) +
  geom_density(fill = "steelblue", alpha = 0.5) +
  geom_vline(xintercept = beta_true[1], colour = "red", linetype = "dashed") +
  labs(title = "VB: beta0", x = "beta0", y = "Density") +
  theme_minimal()

p2 <- ggplot(df_beta1_vb, aes(x = value)) +
  geom_density(fill = "steelblue", alpha = 0.5) +
  geom_vline(xintercept = beta_true[2], colour = "red", linetype = "dashed") +
  labs(title = "VB: beta1", x = "beta1", y = "Density") +
  theme_minimal()

p3 <- ggplot(df_tau_e_vb, aes(x = value)) +
  geom_density(fill = "steelblue", alpha = 0.5) +
  geom_vline(xintercept = tau_e_true, colour = "red", linetype = "dashed") +
  labs(title = "VB: tau_e", x = "tau_e", y = "Density") +
  theme_minimal()

p4 <- ggplot(df_tau_u_vb, aes(x = value)) +
  geom_density(fill = "steelblue", alpha = 0.5) +
  geom_vline(xintercept = tau_u_true, colour = "red", linetype = "dashed") +
  labs(title = "VB: tau_u", x = "tau_u", y = "Density") +
  theme_minimal()

p5 <- ggplot(df_beta0_combined, aes(x = value, fill = method, colour = method)) +
  geom_density(alpha = 0.4) +
  geom_vline(xintercept = beta_true[1], colour = "blue", linetype = "dotted", size = 1.2) +
  scale_fill_manual(values = c(VB = "black", HMC = "#E7298A")) +
  scale_colour_manual(values = c(VB = "black", HMC = "#E7298A")) +
  labs(title = "VB vs HMC: beta0", x = "beta0", y = "Density") +
  theme_minimal()

p6 <- ggplot(df_beta1_combined, aes(x = value, fill = method, colour = method)) +
  geom_density(alpha = 0.4) +
  geom_vline(xintercept = beta_true[2], colour = "blue", linetype = "dotted", size = 1.2) +
  scale_fill_manual(values = c(VB = "black", HMC = "#E7298A")) +
  scale_colour_manual(values = c(VB = "black", HMC = "#E7298A")) +
  labs(title = "VB vs HMC: beta1", x = "beta1", y = "Density") +
  theme_minimal()

p7 <- ggplot(df_tau_e_combined, aes(x = value, fill = method, colour = method)) +
  geom_density(alpha = 0.4) +

```

```

    geom_vline(xintercept = tau_e_true, colour = "blue", linetype = "dotted", size = 1.2) +
    scale_fill_manual(values = c(VB = "black", HMC = "#E7298A")) +
    scale_colour_manual(values = c(VB = "black", HMC = "#E7298A")) +
    labs(title = "VB vs HMC: tau_e", x = "tau_e", y = "Density") +
    theme_minimal()

p8 <- ggplot(df_tauu_combined, aes(x = value, fill = method, colour = method)) +
  geom_density(alpha = 0.4) +
  geom_vline(xintercept = tau_u_true, colour = "blue", linetype = "dotted", size = 1.2) +
  scale_fill_manual(values = c(VB = "black", HMC = "#E7298A")) +
  scale_colour_manual(values = c(VB = "black", HMC = "#E7298A")) +
  labs(title = "VB vs HMC: tau_u", x = "tau_u", y = "Density") +
  theme_minimal()

# Combine all 8 panels
combined_plot <- (p1 + p2 + p3 + p4) / (p5 + p6 + p7 + p8)
plot_filename <- glue("{model_prefix}vb_Q{q_temp}_8Panel.png")

ggsave(
  filename = glue("../figs/{plot_filename}"),
  plot      = combined_plot,
  width     = 16,
  height    = 10,
  dpi       = 300
)

cat(glue("Generated plots for Q={q_temp}: {scenario_label}_convergence.png, {scenario_label}_elbo.png"))
}

```

```

##
## --- Running: Q=5 (n=60 per group) ---
## Running Gibbs with 3 different initial values...
## Combined 12000 samples from 3 Gibbs chains
## VB E[tau_u]: 1.1115
## HMC E[tau_u]: 1.1562

## Generated plots for Q=5: s1_convergence.png, s1_elbo.png, M0_vb_Q5_8Panel.png
## --- Running: Q=10 (n=30 per group) ---
## Running Gibbs with 3 different initial values...
## Combined 12000 samples from 3 Gibbs chains
## VB E[tau_u]: 0.9441
## HMC E[tau_u]: 0.9802

## Generated plots for Q=10: s2_convergence.png, s2_elbo.png, M0_vb_Q10_8Panel.png
## --- Running: Q=20 (n=15 per group) ---
## Running Gibbs with 3 different initial values...
## Combined 12000 samples from 3 Gibbs chains
## VB E[tau_u]: 1.1515
## HMC E[tau_u]: 1.2023

## Generated plots for Q=20: s3_convergence.png, s3_elbo.png, M0_vb_Q20_8Panel.png
## --- Running: Q=50 (n=6 per group) ---
## Running Gibbs with 3 different initial values...
## Combined 12000 samples from 3 Gibbs chains
## VB E[tau_u]: 0.9933

```

```

## HMC E[tau_u]: 1.0404
## Generated plots for Q=50: s4_convergence.png, s4_elbo.png, M0_vb_Q50_8Panel.png
## --- Running: Q=100 (n=3 per group) ---
## Running Gibbs with 3 different initial values...
## Combined 12000 samples from 3 Gibbs chains
## VB E[tau_u]: 1.6673
## HMC E[tau_u]: 2.2709
## Generated plots for Q=100: s5_convergence.png, s5_elbo.png, M0_vb_Q100_8Panel.png
cat("\n===== \n")

##
## =====

```

2 Aggregate Analysis Across Configurations

```

# Check if history data is available
if (!is.null(results_multi[[1]]$vb$E_tau_u_history) && length(results_multi[[1]]$vb$E_tau_u_history) > 0) {
  par(mfrow = c(1, 2))

  # E[tau_u] convergence for all 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)))

  for (i in 2:length(results_multi)) {
    lines(1:length(results_multi[[i]]$vb$E_tau_u_history),
          results_multi[[i]]$vb$E_tau_u_history,
          col = c('#7570b3', '#e7298a', '#d95f02', '#66a61e')[i-1],
          lwd = 2)
  }

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

  legend('topright',
        legend = c('Q=5', 'Q=10', 'Q=20', 'Q=50', 'Q=100', 'True value'),
        col = c('#1b9e77', '#7570b3', '#e7298a', '#d95f02', '#66a61e', 'red'),
        lty = c(rep(1, 5), 2),
        lwd = 2,
        cex = 0.8)

  # ELBO convergence for all 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)))

  for (i in 2:length(results_multi)) {
    lines(1:length(results_multi[[i]]$vb$elbo_history),

```

```

      results_multi[[i]]$vb$elbo_history,
      col = c('#7570b3', '#e7298a', '#d95f02', '#66a61e')[i-1],
      lwd = 2)
}

legend('bottomright',
      legend = c('Q=5', 'Q=10', 'Q=20', 'Q=50', 'Q=100'),
      col = c('#1b9e77', '#7570b3', '#e7298a', '#d95f02', '#66a61e'),
      lty = 1,
      lwd = 2,
      cex = 0.8)
grid()
} else {
  plot.new()
  text(0.5, 0.5, "Convergence history not available\n(Using Dr John's original functions)",
       cex = 1.5, col = "gray50")
}

```

Convergence history not available
(Using Dr John's original functions)

```

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

  x_min <- min(vb_min, gibbs_min, tau_u_true * 0.5)
  x_max <- max(vb_max, gibbs_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, colour = method, linetype = method)) +
  geom_line(linewidth = 1.2) +

```

```

    geom_vline(xintercept = tau_u_true, colour = "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 all 5 plots into a grid (2 rows, 3 columns)
  combined_tau_u <- (plot_list[[1]] | plot_list[[2]] | plot_list[[3]]) /
    (plot_list[[4]] | plot_list[[5]] | plot_spacer()) +
    plot_annotation(
      title = "Effect of Sample Size Per Group on _u Posterior",
      subtitle = "VB approximation quality with sufficient groups for variance component estimation",
      theme = theme(
        plot.title = element_text(size = 16, face = "bold"),
        plot.subtitle = element_text(size = 12)
      )
    )

  # Save plot
  ggsave(
    filename = glue("../figs/{model_prefix}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")

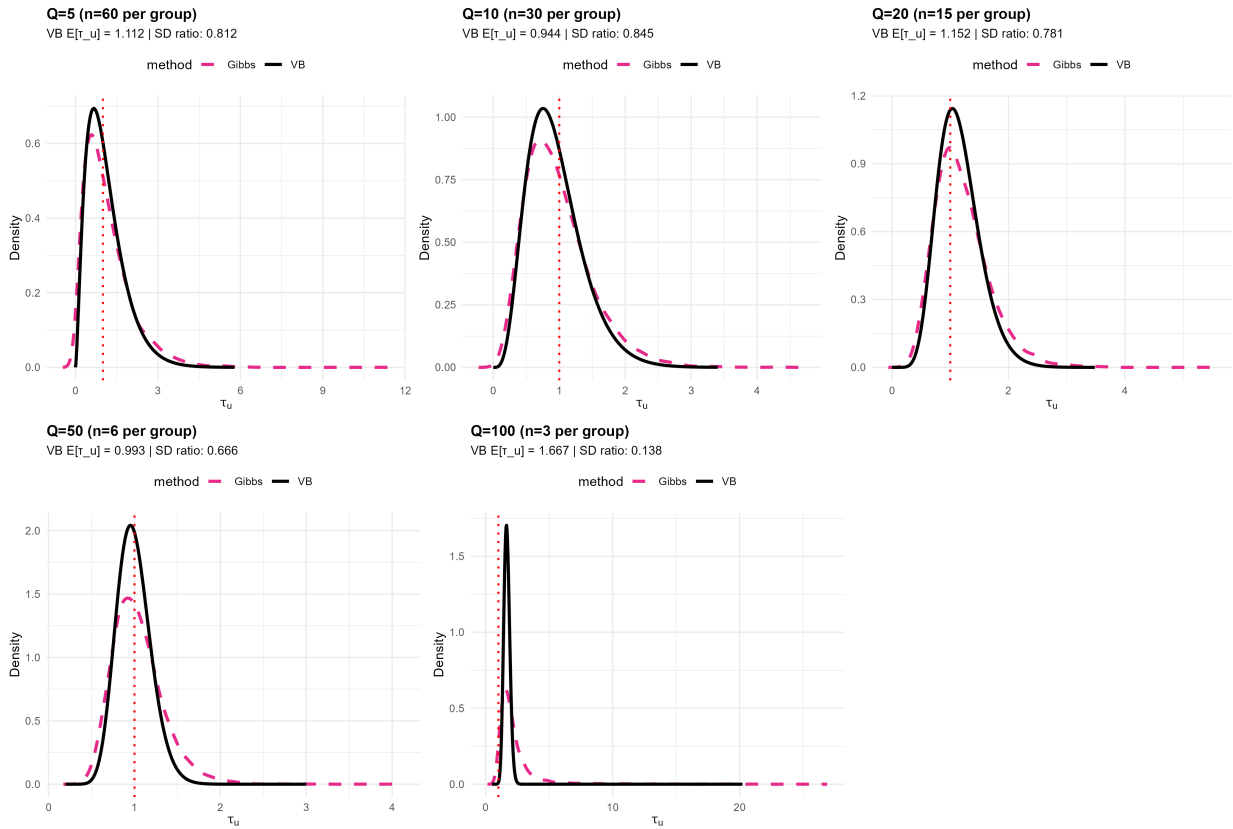
## _u comparison plot saved to figs/tau_u_sample_size_comparison.png

# Display
img_tau_u <- readPNG(glue("../figs/{model_prefix}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 quality with sufficient groups for variance component estimation



```
# 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, colour = config)) +
    geom_line(linewidth = 1.2) +
```



```

geom_vline(xintercept = tau_u_true, colour = "red", linetype = "dotted", linewidth = 0.8) +
scale_color_manual(
  values = c(
    "Q=5 (n=60 per group)" = "gray50",
    "Q=10 (n=30 per group)" = "#d95f02",
    "Q=20 (n=15 per group)" = "#7570b3",
    "Q=50 (n=6 per group)" = "#e7298a",
    "Q=100 (n=3 per group)" = "#1b9e77"
  )
) +
coord_cartesian(xlim = c(0, 8)) +
labs(
  title = "Gibbs Sampling Posteriors",
  subtitle = "All configurations show similar distributions",
  x = expression(tau[u]),
  y = "Density",
  colour = "Configuration"
) +
theme_minimal() +
theme(
  legend.position = "right",
  legend.text = element_text(size = 9),
  legend.title = element_text(size = 10, face = "bold"),
  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, colour = config)) +

```

```

geom_line(linewidth = 1.2) +
geom_vline(xintercept = tau_u_true, colour = "red", linetype = "dotted", linewidth = 0.8) +
scale_color_manual(
  values = c(
    "Q=5 (n=60 per group)" = "gray50",
    "Q=10 (n=30 per group)" = "#d95f02",
    "Q=20 (n=15 per group)" = "#7570b3",
    "Q=50 (n=6 per group)" = "#e7298a",
    "Q=100 (n=3 per group)" = "#1b9e77"
  )
) +
coord_cartesian(xlim = c(0, 8)) +
labs(
  title = "VB Posteriors",
  subtitle = "Consistent performance with sufficient groups",
  x = expression(tau[u]),
  y = "Density",
  colour = "Configuration"
) +
theme_minimal() +
theme(
  legend.position = "right",
  legend.text = element_text(size = 9),
  legend.title = element_text(size = 10, face = "bold"),
  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 p"
  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 = glue("../figs/{model_prefix}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")

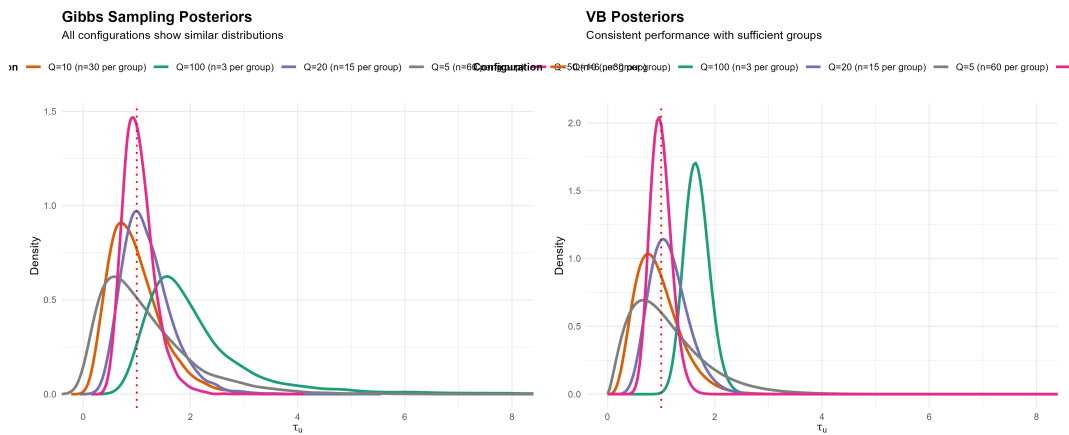
## _u overlay comparison plot saved to figs/tau_u_overlay_comparison.png

# Display
img_overlay <- readPNG(glue("../figs/{model_prefix}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



```

# Generate MY plot in Dr John's exact style
# Single panel with all Gibbs (solid) and VB (dashed) overlaid

```

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

```

##
## =====
cat("My _u Comparison (Dr John's Style)\n")

```

```
## My _u Comparison (Dr John's Style)
```

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

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

```

# Prepare data: combine all Gibbs chains for each Q
gibbs_combined <- list()
vb_params <- list()

for (i in seq_along(results_multi)) {
  cfg <- group_configs[[i]]
  q_val <- cfg$q

  if (run_gibbs) {
    # results_multi[[i]]$gibbs is a matrix from run_gibbs_sampler
    # tau_u column name is "tau_u"
    gibbs_matrix <- results_multi[[i]]$gibbs
    gibbs_combined[[paste0("q", q_val)]] <- gibbs_matrix[, "tau_u"]
  }

  # VB gamma parameters from results_multi
  vb_result <- results_multi[[i]]$vb
  a_param <- vb_result$a_u_new
  b_param <- vb_result$b_u_new
  vb_params[[paste0("q", q_val)]] <- list(a = a_param, b = b_param)
}

# Create plot matching Dr John's style
png(filename = "../figs/my_tau_u_comparison.png", width = 10, height = 8, units = "in", res = 300)

# Colors for all 5 Q values
colors <- c("black", "red", "green3", "blue", "purple")
q_values <- c(5, 10, 20, 50, 100)

# Start with first Gibbs density
if (run_gibbs) {
  plot(density(gibbs_combined$q5),
       xlab = expression(tau['u']),
       main = '',
       ylim = c(0, 2.5),
       xlim = c(0, 8),
       lwd = 2,
       col = colors[1])

  # Add remaining Gibbs densities
  lines(density(gibbs_combined$q10), col = colors[2], lwd = 2)
  lines(density(gibbs_combined$q20), col = colors[3], lwd = 2)
  lines(density(gibbs_combined$q50), col = colors[4], lwd = 2)
  lines(density(gibbs_combined$q100), col = colors[5], lwd = 2)

  # Add VB approximations (dashed)
  curve(dgamma(x, vb_params$q5$a, vb_params$q5$b),
        add = TRUE, lty = 2, lwd = 2, col = colors[1])
  curve(dgamma(x, vb_params$q10$a, vb_params$q10$b),
        add = TRUE, lty = 2, lwd = 2, col = colors[2])
  curve(dgamma(x, vb_params$q20$a, vb_params$q20$b),
        add = TRUE, lty = 2, lwd = 2, col = colors[3])
  curve(dgamma(x, vb_params$q50$a, vb_params$q50$b),

```

```

      add = TRUE, lty = 2, lwd = 2, col = colors[4])
curve(dgamma(x, vb_params$q100$a, vb_params$q100$b),
      add = TRUE, lty = 2, lwd = 2, col = colors[5])

# Legend
legend('topright',
      col = c(colors, "black", "black"),
      lty = c(0, 0, 0, 0, 0, 1, 2),
      lwd = c(NA, NA, NA, NA, NA, 2, 2),
      pch = c(19, 19, 19, 19, 19, NA, NA),
      legend = c('q=5', 'q=10', 'q=20', 'q=50', 'q=100', 'Posterior (Gibbs)', 'VB approximation'),
      cex = 0.9)
} else {
  # VB only version
  curve(dgamma(x, vb_params$q5$a, vb_params$q5$b),
        from = 0, to = 8,
        xlab = expression(tau['u']),
        ylab = "Density",
        main = '',
        ylim = c(0, 2.5),
        lwd = 2,
        col = colors[1])

  curve(dgamma(x, vb_params$q10$a, vb_params$q10$b),
        add = TRUE, lwd = 2, col = colors[2])
  curve(dgamma(x, vb_params$q20$a, vb_params$q20$b),
        add = TRUE, lwd = 2, col = colors[3])
  curve(dgamma(x, vb_params$q50$a, vb_params$q50$b),
        add = TRUE, lwd = 2, col = colors[4])
  curve(dgamma(x, vb_params$q100$a, vb_params$q100$b),
        add = TRUE, lwd = 2, col = colors[5])

  legend('topright',
        col = colors,
        lty = 1,
        lwd = 2,
        legend = c('q=5', 'q=10', 'q=20', 'q=50', 'q=100'),
        cex = 0.9)
}

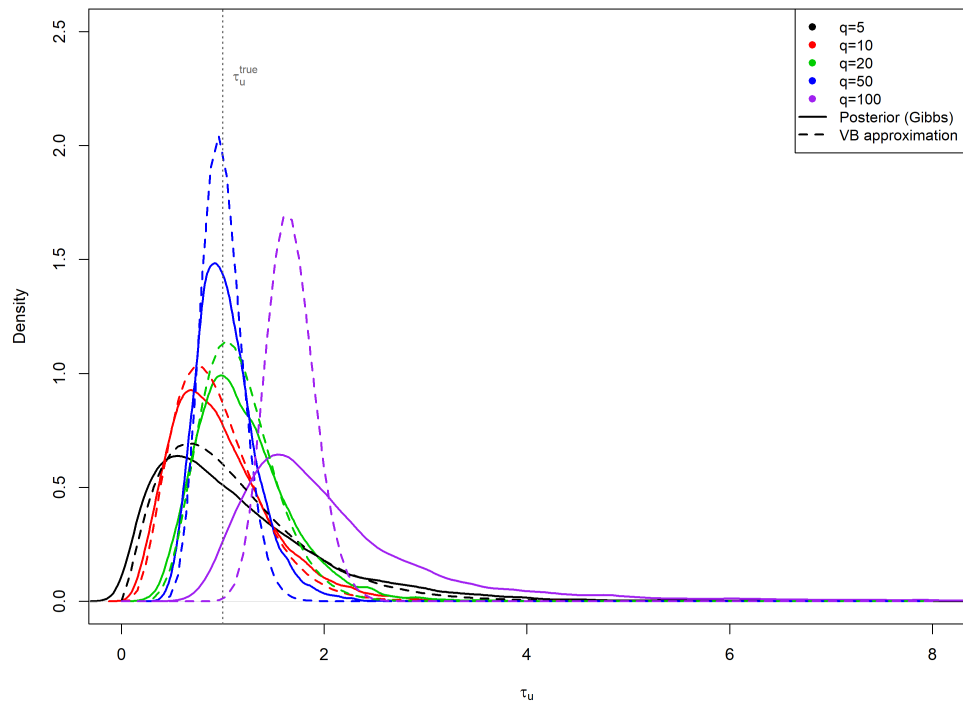
# Add vertical line at true value
abline(v = tau_u_true, lty = 3, col = "gray40", lwd = 1.5)
text(tau_u_true, 2.3, labels = expression(tau[u]^true), pos = 4, cex = 0.9, col = "gray40")

dev.off()

## pdf
## 2
cat("My plot saved to figs/my_tau_u_comparison.png\n")

## My plot saved to figs/my_tau_u_comparison.png

```



```
# Load Dr John's reference results from RDS
cat("\n===== \n")

##
## =====

cat("Loading Dr John's Reference Data (RDS)\n")

## Loading Dr John's Reference Data (RDS)
cat("===== \n\n")

## =====

# Try relative path first, then fall back to absolute
rds_path <- "../results/dr_john_reference_tau_u.rds"
if (!file.exists(rds_path)) {
  rds_path <- "d:/github/VI1/results/dr_john_reference_tau_u.rds"
}

if (file.exists(rds_path)) {
  cat("Loading:", rds_path, "\n")
  dr_john_ref <- readRDS(rds_path)

  # Check structure
  cat("\nVB structure check:\n")
  cat("  q5 VB class:", class(dr_john_ref$vb$q5), "\n")
  if (is.list(dr_john_ref$vb$q5)) {
    cat("  q5 VB names:", names(dr_john_ref$vb$q5), "\n")
    cat("  q5$a =", dr_john_ref$vb$q5$a, "  q5$b =", dr_john_ref$vb$q5$b, "\n")
  } else {
```

```

  cat("  q5 VB values:", dr_john_ref$vb$q5, "\n")
}

cat("\nCreating plot from Dr John's saved RDS data...\n")

colors_rds <- c("black", "red", "green3", "blue", "magenta")

# Save plot to PNG file (try relative path first, then absolute)
png_path <- "../figs/my_from_rds_tau_u_comparison.png"
if (!dir.exists("../figs")) png_path <- "d:/github/VI1/figs/my_from_rds_tau_u_comparison.png"
png(filename = png_path, width = 3500, height = 2800, res = 300)

plot(density(dr_john_ref$gibbs$q5),
      xlab=expression(tau['u']),
      main='Recreated from Dr John\'s RDS File',
      ylim=c(0, 2.5),
      xlim=c(0, 8),
      lwd=2,
      col=colors_rds[1])

lines(density(dr_john_ref$gibbs$q10), col=colors_rds[2], lwd=2)
lines(density(dr_john_ref$gibbs$q20), col=colors_rds[3], lwd=2)
lines(density(dr_john_ref$gibbs$q50), col=colors_rds[4], lwd=2)
lines(density(dr_john_ref$gibbs$q100), col=colors_rds[5], lwd=2)

# Add VB approximations - check if data is vector or list
if (is.list(dr_john_ref$vb$q5)) {
  # VB stored as list(a=, b=)
  curve(dgamma(x, dr_john_ref$vb$q5$a, dr_john_ref$vb$q5$b),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[1])
  curve(dgamma(x, dr_john_ref$vb$q10$a, dr_john_ref$vb$q10$b),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[2])
  curve(dgamma(x, dr_john_ref$vb$q20$a, dr_john_ref$vb$q20$b),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[3])
  curve(dgamma(x, dr_john_ref$vb$q50$a, dr_john_ref$vb$q50$b),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[4])
  curve(dgamma(x, dr_john_ref$vb$q100$a, dr_john_ref$vb$q100$b),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[5])
} else {
  # VB stored as vector c(a, b)
  curve(dgamma(x, dr_john_ref$vb$q5[1], dr_john_ref$vb$q5[2]),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[1])
  curve(dgamma(x, dr_john_ref$vb$q10[1], dr_john_ref$vb$q10[2]),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[2])
  curve(dgamma(x, dr_john_ref$vb$q20[1], dr_john_ref$vb$q20[2]),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[3])
  curve(dgamma(x, dr_john_ref$vb$q50[1], dr_john_ref$vb$q50[2]),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[4])
  curve(dgamma(x, dr_john_ref$vb$q100[1], dr_john_ref$vb$q100[2]),
        add = TRUE, lty = 2, lwd = 2, col = colors_rds[5])
}

legend('topright',

```

```

col=c(colors_rds, "black", "black"),
lty=c(0,0,0,0,0,1,2),
lwd=c(NA,NA,NA,NA,NA,2,2),
pch=c(19,19,19,19,19,NA,NA),
legend=c('q=5', 'q=10', 'q=20', 'q=50', 'q=100', 'Posterior (Gibbs)', 'VB approximation'),
cex=0.9)

abline(v=0.5, lty=3, col="gray40", lwd=1.5)
text(0.5, 2.3, labels=expression(tau[u]^true* = 0.5), pos=4, cex=0.9, col="gray40")

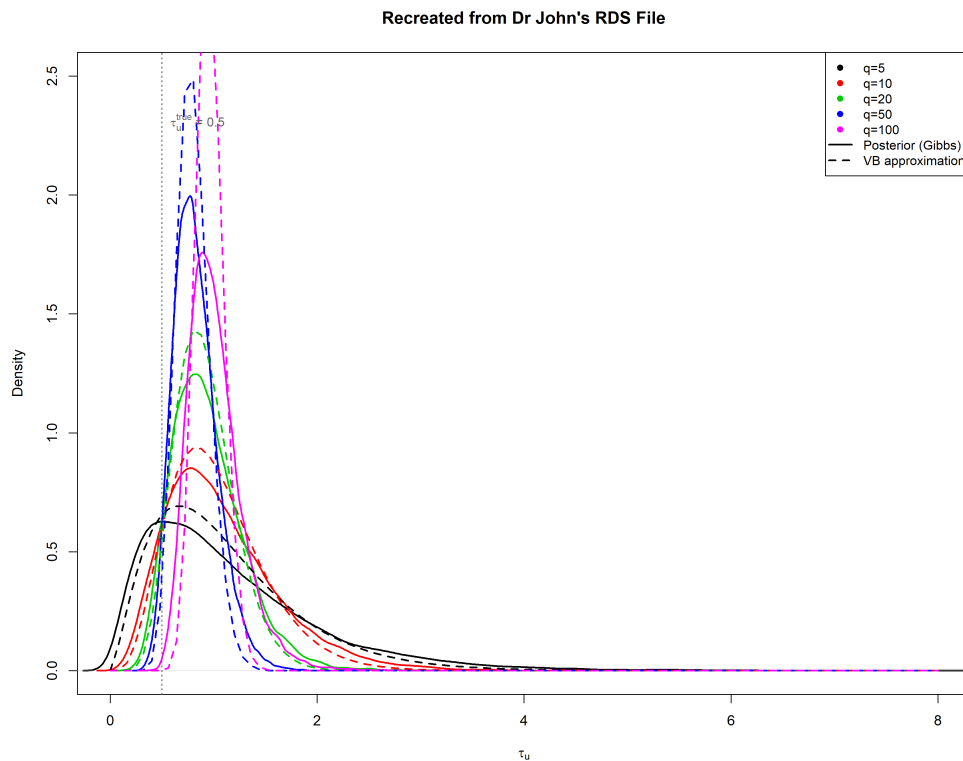
dev.off()

cat("Plot from RDS saved to figs/my_from_rds_tau_u_comparison.png\n")

} else {
cat("WARNING: RDS file not found at:", rds_path, "\n")
cat("Run 'Code for David Ewing Random intercept example.R' to generate it.\n")
}

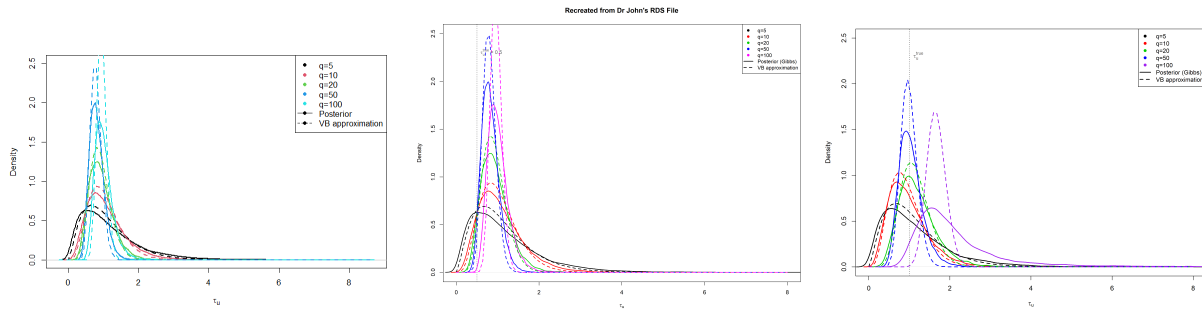
## Loading: ../results/dr_john_reference_tau_u.rds
##
## VB structure check:
##   q5 VB class: numeric
##   q5 VB values: 2.5 2.249205
##
## Creating plot from Dr John's saved RDS data...
## Plot from RDS saved to figs/my_from_rds_tau_u_comparison.png

```



2.1 Three-Panel Validation Comparison

Panel status: - Dr John's baseline PNG: - My plot from RDS: - My new run with matched params:



Left: Dr John's baseline PNG (from his .R file)

Centre: My recreation from his RDS data

Right: My new run with matched parameters

```
# 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("\nDiagnostic: Var_posterior(u) / Var_prior(u)\n")
```

```
##
```

```
## Diagnostic: Var_posterior(u) / Var_prior(u)
```

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

```
## Diagnostic: Var_posterior(u) / Var_prior(u)
```

```
# 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 0.2052288  0.3772637 Q=5 (n=60 per group)
## 2   10           30 0.1628947  0.1939433 Q=10 (n=30 per group)
## 3   20           15 0.1542528  0.1615703 Q=20 (n=15 per group)
## 4   50            6 0.2766444  0.2810183 Q=50 (n=6 per group)
## 5  100            3 0.3415518  0.3315218 Q=100 (n=3 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$Gibbs <- ratio_data$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, colour = 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))",
  colour = "Method"
) +
theme_minimal() +
theme(
  legend.position = "top",
  plot.title = element_text(size = 14, face = "bold"),
  plot.subtitle = element_text(size = 11)
)

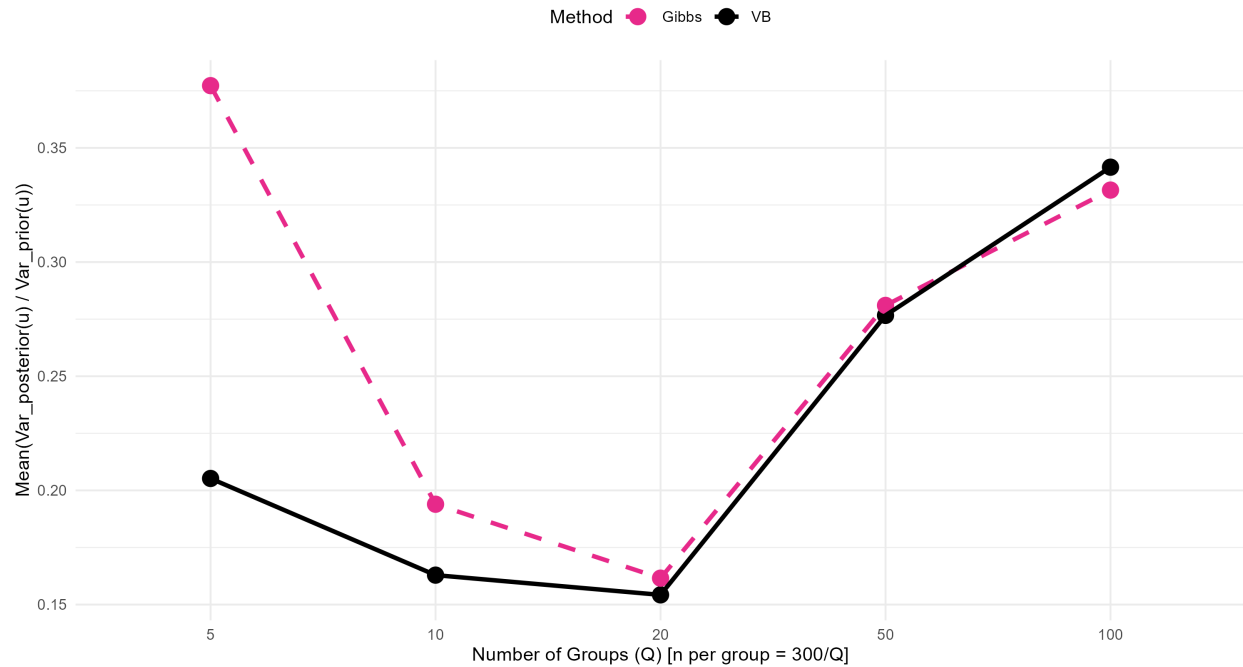
ggsave(
  filename = glue("../figs/{model_prefix}diagnostic_variance_ratio.png"),
  plot = p_diagnostic,
  width = 10,
  height = 6,
  dpi = 300
)

img_diagnostic <- readPNG(glue("../figs/{model_prefix}diagnostic_variance_ratio.png"))
grid.newpage()
grid.raster(img_diagnostic)

```

Diagnostic: Posterior Variance / Prior Variance Ratio for Random Effects

Varying Q (fixed N=300): Lower ratio indicates concentrated posteriors and better τ_u estimation



```
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")
```

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

```
##
```

```
## Saved MO SD ratios for 5 configurations
```