Variational Bayes: DEBUG3 - Testing Prior Strength (Weak vs Strong)

2026-02-06 17:56:46

Contents

```r
# ============================================================================
# MODEL TYPE SELECTION
# M1: Linear regression with   and  ² (residual variance)
#     - No random effects, no variance component

# M3: Hierarchical model with  , u (random effects),  ,
#     - 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 - SCENARIO 1 (WEAK PRIOR)
alpha_e_s1 <- 0.01
gamma_e_s1 <- 0.01
alpha_u_s1 <- 1.0   # Weak: Var = 1/4 = 0.25 (wide)
gamma_u_s1 <- 2.0   # Mean = 1.0/2.0 = 0.5

# Prior Hyperparameters - SCENARIO 2 (STRONG PRIOR)
alpha_e_s2 <- 0.01
gamma_e_s2 <- 0.01
alpha_u_s2 <- 5.0   # Strong: Var = 5/100 = 0.05 (tight)
gamma_u_s2 <- 10.0  # Mean = 5.0/10.0 = 0.5

# Gibbs Sampler Settings
run_gibbs    <- TRUE   # Set FALSE for quick testing, TRUE for full run
gibbs_iter   <- 5000
gibbs_burnin <- 1000
```

```r
# Display Settings
RENDER_FUNCTIONS <- FALSE  # TRUE to show function code, FALSE to hide

# SCENARIO 1: 6 groups with WEAK prior - M3 only
q_30levels  <- if (model_type == "M1") 0 else 6
nq_30levels <- if (model_type == "M1") 0 else 50

# SCENARIO 2: 6 groups with STRONG prior - M3 only
q_6levels  <- if (model_type == "M1") 0 else 6
nq_6levels <- if (model_type == "M1") 0 else 50

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

1 Scenario 1: Conditional on Model Type

```
cat(glue("\n=== Running {model_type} Analysis ===\n"))
```

```
## === Running M3 Analysis ===
```

```
q_s1  <- q_30levels
nq_s1 <- nq_30levels

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") "SCENARIO 1: 6 levels, WEAK prior (alpha=1, gamma=2)" else "SC
cat(glue("\n=== {scenario_name} ===\n"))
```

```
## === SCENARIO 1: 6 levels, WEAK prior (alpha=1, gamma=2) ===
```

```
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_s1,
  gamma_e    = gamma_e_s1,
  alpha_u    = alpha_u_s1,
  gamma_u    = gamma_u_s1,
  model_type = model_type
)
```

```
##
## === Iteration 1 ===
##   E_tau_e=1.000000, E_tau_u=0.500000
##   mu_betau[1:3]: [4.9891, 0.4824, 0.4615]
##   Sigma_betau[1,1]=0.336813, Sigma_betau[p+1,p+1]=0.350012
##   mu_u: [0.9511, -0.5783, 0.4552, 0.1563, -1.3517, 0.3674]
##   ||mu_u||^2 = 3.432654
##   quad_form = mu_u' * K_inv * mu_u = 3.432654
##   trace_u = tr(K_inv * Sigma_uu) = 2.100442
##   a_u_new = 4.0000, b_u_new = 4.7665
##   E[tau_u] = a_u/b_u = 0.839182
##   Change in E[tau_u]: 0.339182
##
## === Iteration 2 ===
##   E_tau_e=4.443423, E_tau_u=0.839182
##   mu_betau[1:3]: [4.9891, 0.4824, 0.4611]
##   Sigma_betau[1,1]=0.199389, Sigma_betau[p+1,p+1]=0.202383
##   mu_u: [0.9570, -0.5819, 0.4581, 0.1572, -1.3601, 0.3697]
##   ||mu_u||^2 = 3.475608
##   quad_form = mu_u' * K_inv * mu_u = 3.475608
##   trace_u = tr(K_inv * Sigma_uu) = 1.214384
##   a_u_new = 4.0000, b_u_new = 4.3450
##   E[tau_u] = a_u/b_u = 0.920599
##   Change in E[tau_u]: 0.081418
##
## === Iteration 3 ===
##   E_tau_e=4.952735, E_tau_u=0.920599
##   mu_betau[1:3]: [4.9891, 0.4824, 0.4611]
##   Sigma_betau[1,1]=0.181744, Sigma_betau[p+1,p+1]=0.184430
##   mu_u: [0.9571, -0.5819, 0.4581, 0.1572, -1.3602, 0.3698]
##   ||mu_u||^2 = 3.476023
##   quad_form = mu_u' * K_inv * mu_u = 3.476023
##   trace_u = tr(K_inv * Sigma_uu) = 1.106658
##   a_u_new = 4.0000, b_u_new = 4.2913
##   E[tau_u] = a_u/b_u = 0.932110
##   Change in E[tau_u]: 0.011510
## Converged at iteration 8
## Max relative change: 3.17e-06
```

```r
exact_s1 <- exact_linear_posterior(
  X           = X_s1,
  y           = y_s1,
  prior_mu    = rep(0, p),
  prior_sigma = 10,
  prior_a     = alpha_e_s1,
  prior_b     = gamma_e_s1
)

cat("\nExact posterior parameters:\n")
```

```
##
## Exact posterior parameters:
```

```r
cat("E[beta]:", exact_s1$mu_n, "\n")
```

```
## E[beta]: 4.996952 0.4723502 0.5143735 0.4958919
```

```r
cat("E[tau_e]:", exact_s1$a_n / exact_s1$b_n, "\n")
```

```
## E[tau_e]: 1.288377
```

```r
cat("\nRunning Laplace approximation...\n")
```

```
##
## Running Laplace approximation...
```

```
laplace_s1 <- run_laplace(
  X          = X_s1,
  Z          = Z_s1,
  y          = y_s1,
  p          = p,
  q          = q_s1,
  n          = n,
  alpha_e    = alpha_e_s1,
  gamma_e    = gamma_e_s1,
  alpha_u    = alpha_u_s1,
  gamma_u    = gamma_u_s1,
  model_type = model_type
)

cat("Laplace MAP estimates:\n")
```

```
## Laplace MAP estimates:
```

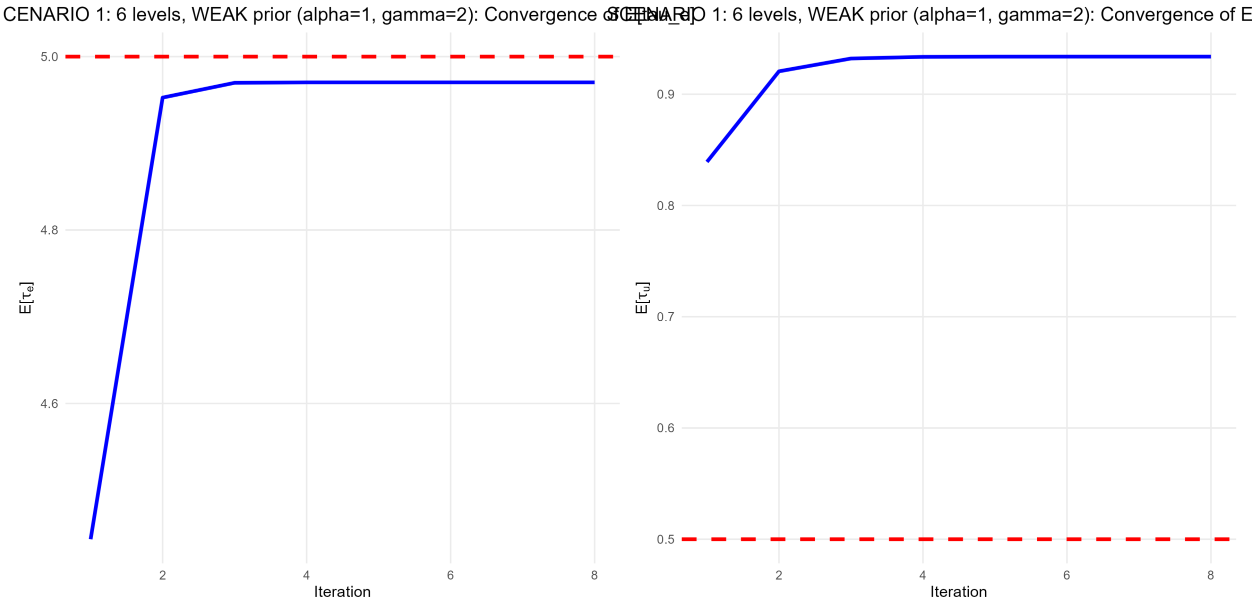
```
if (model_type == "M1") {
  cat("beta:", laplace_s1$theta_map[1:p], "\n")
  cat("tau_e:", exp(laplace_s1$theta_map[p + 1]), "\n")
} else {
  cat("beta:", laplace_s1$theta_map[1:p], "\n")
  cat("tau_e:", exp(laplace_s1$theta_map[p + q_s1 + 1]), "\n")
  cat("tau_u:", exp(laplace_s1$theta_map[p + q_s1 + 2]), "\n")
}
```

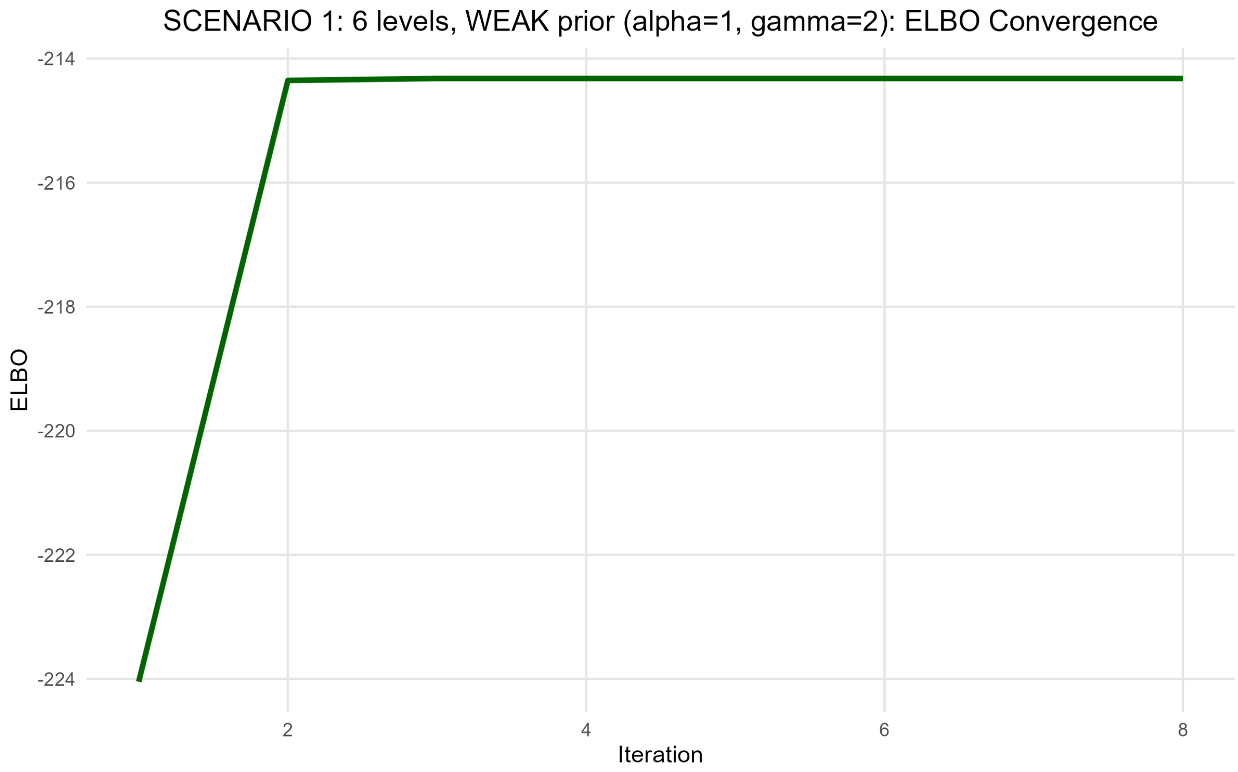
```
## beta: 4.978784 0.4824338 0.4610824 0.4650453
## tau_e: 5.089609
## tau_u: 0.8021855
```

```
# Save results for VI-COMPARE
if (model_type == "M3") {
  saveRDS(list(
    vb = results_s1,
    laplace = laplace_s1,
    data = list(X = X_s1, Z = Z_s1, y = y_s1),
    truth = list(tau_u = tau_u_true, tau_e = tau_e_true),
    config = list(q = q_s1, nq = nq_30levels,
                  alpha_u = alpha_u_s1, gamma_u = gamma_u_s1,
                  prior_type = "weak")
  ), file.path("..", "results", "debug3_s1_weak.rds"))
}
```

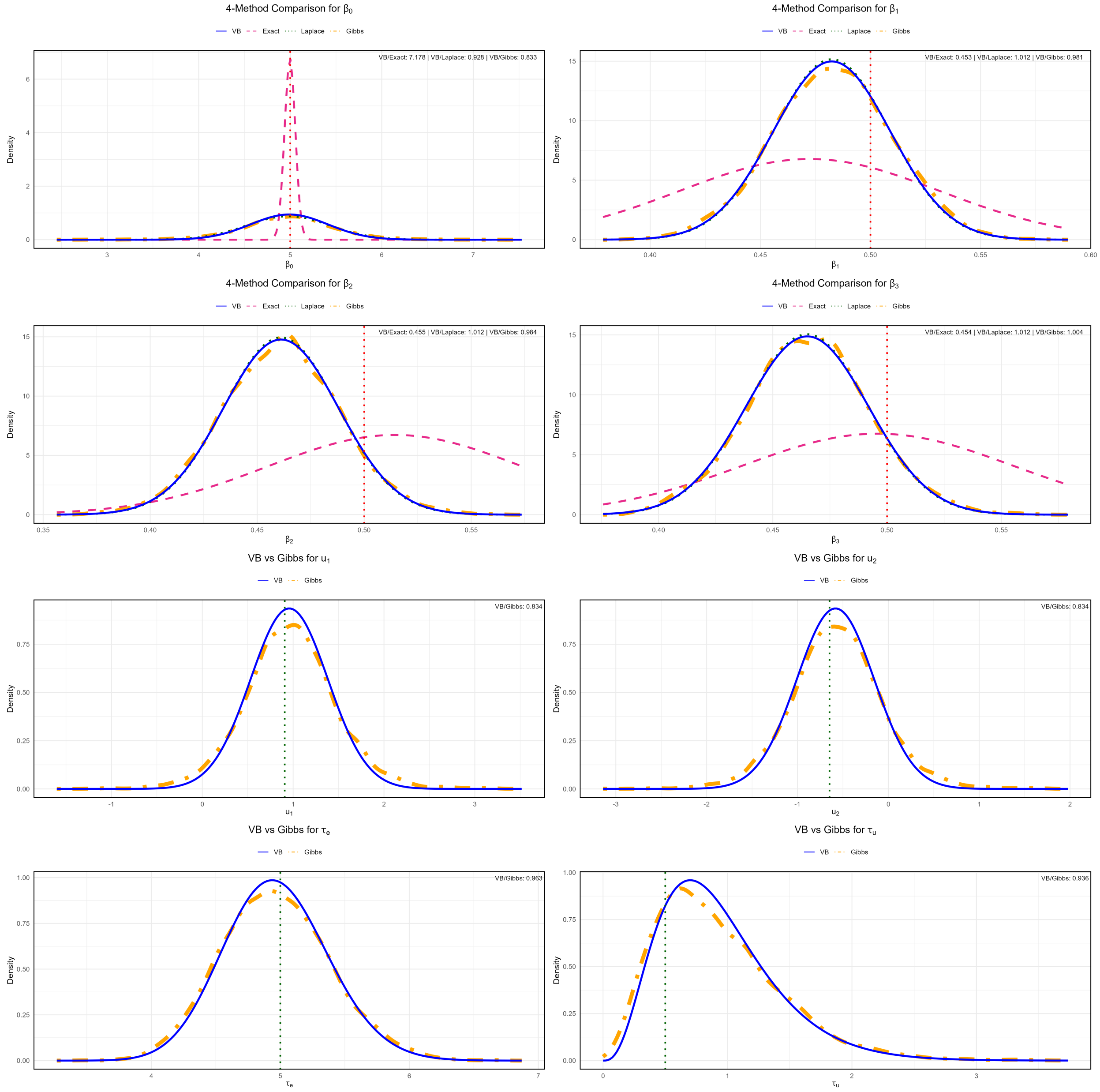
```
if (run_gibbs) {
  cat("\nRunning Gibbs sampler...\n")
  gibbs_s1 <- run_gibbs_sampler(
    X          = X_s1,
    Z          = Z_s1,
    y          = y_s1,
    p          = p,
    q          = q_s1,
    n          = n,
    alpha_e    = alpha_e_s1,
    gamma_e    = gamma_e_s1,
    alpha_u    = alpha_u_s1,
    gamma_u    = gamma_u_s1,
    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: 4.97638 0.4826977 0.4608714 0.4655507
## tau_e: 4.964147
## tau_u: 0.9273652
```





```
## M3: 8-panel ggplot saved
```


4-Method Comparison for β_0

4-Method Comparison for β_1

4-Method Comparison for β_2

4-Method Comparison for β_3

VB vs Gibbs for u_1

VB vs Gibbs for u_2

VB vs Gibbs for τ_e

VB vs Gibbs for τ_u

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

```r
if (model_type == "M3") {
  q_s2      <- q_6levels
  nq_s2     <- nq_6levels
  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)

  cat("\n=== DEBUG: Scenario 2 Data Generation ===\n")
  cat(sprintf("True tau_u = %.4f, True tau_e = %.4f\n", tau_u_true, tau_e_true))
  cat(sprintf("True random effects u: mean=%.4f, sd=%.4f\n", mean(u_true_s2), sd(u_true_s2)))
  cat(sprintf("Range of u: [%.4f, %.4f]\n", min(u_true_s2), max(u_true_s2)))
  cat(sprintf("Empirical var(u) = %.4f, theoretical = %.4f\n", var(u_true_s2), 1/tau_u_true))
} else {
  cat("\nScenario 2 skipped (M3 only)\n")
}
```

```
##
## === DEBUG: Scenario 2 Data Generation ===
## True tau_u = 0.5000, True tau_e = 5.0000
## True random effects u: mean=-0.0913, sd=0.5797
## Range of u: [-0.7282, 0.8065]
## Empirical var(u) = 0.3361, theoretical = 2.0000
```

```r
cat("\n=== SCENARIO 2: 6 levels, STRONG prior (alpha=5, gamma=10) ===\n")
```

```
##
## === SCENARIO 2: 6 levels, STRONG prior (alpha=5, gamma=10) ===
```

```r
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_s2,
  gamma_e    = gamma_e_s2,
  alpha_u    = alpha_u_s2,
  gamma_u    = gamma_u_s2,
  model_type = model_type
)
```

```
##
## === Iteration 1 ===
##   E_tau_e=1.000000, E_tau_u=0.500000
##   mu_betau[1:3]: [4.9177, 0.4732, 0.5170]
##   Sigma_betau[1,1]=0.336732, Sigma_betau[p+1,p+1]=0.349907
##   mu_u: [-0.5170, 0.3748, 0.8475, -0.3527, -0.3464, -0.0063]
##   ||mu_u||^2 = 1.370392
##   quad_form = mu_u' * K_inv * mu_u = 1.370392
##   trace_u = tr(K_inv * Sigma_uu) = 2.099931
##   a_u_new = 8.0000, b_u_new = 11.7352
##   E[tau_u] = a_u/b_u = 0.681712
##   Change in E[tau_u]: 0.181712
##
## === Iteration 2 ===
##   E_tau_e=4.526960, E_tau_u=0.681712
##   mu_betau[1:3]: [4.9177, 0.4733, 0.5169]
##   Sigma_betau[1,1]=0.245233, Sigma_betau[p+1,p+1]=0.248169
##   mu_u: [-0.5206, 0.3774, 0.8535, -0.3552, -0.3488, -0.0063]
##   ||mu_u||^2 = 1.389705
##   quad_form = mu_u' * K_inv * mu_u = 1.389705
##   trace_u = tr(K_inv * Sigma_uu) = 1.489125
##   a_u_new = 8.0000, b_u_new = 11.4394
##   E[tau_u] = a_u/b_u = 0.699336
##   Change in E[tau_u]: 0.017624
##
## === Iteration 3 ===
##   E_tau_e=5.059147, E_tau_u=0.699336
##   mu_betau[1:3]: [4.9177, 0.4733, 0.5169]
##   Sigma_betau[1,1]=0.238993, Sigma_betau[p+1,p+1]=0.241621
##   mu_u: [-0.5207, 0.3775, 0.8537, -0.3553, -0.3489, -0.0063]
##   ||mu_u||^2 = 1.390396
##   quad_form = mu_u' * K_inv * mu_u = 1.390396
##   trace_u = tr(K_inv * Sigma_uu) = 1.449823
##   a_u_new = 8.0000, b_u_new = 11.4201
##   E[tau_u] = a_u/b_u = 0.700519
##   Change in E[tau_u]: 0.001182
## Converged at iteration 6
## Max relative change: 6.56e-06
```

```r
exact_s2 <- exact_linear_posterior(
  X           = X_s2,
  y           = y_s2,
  prior_mu    = rep(0, p),
  prior_sigma = 10,
  prior_a     = alpha_e_s2,
  prior_b     = gamma_e_s2
)

cat("\nExact posterior parameters:\n")
```

```
##
## Exact posterior parameters:
```

```r
cat("E[beta]:", exact_s2$mu_n, "\n")
```

```
## E[beta]: 4.919998 0.4612319 0.5299647 0.5829415
```

```r
cat("E[tau_e]:", exact_s2$a_n / exact_s2$b_n, "\n")
```

```
## E[tau_e]: 2.363604
```

```r
cat("\nRunning Laplace approximation...\n")
```

```
##
## Running Laplace approximation...
```

```r
laplace_s2 <- run_laplace(
  X          = X_s2,
  Z          = Z_s2,
  y          = y_s2,
  p          = p,
  q          = q_s2,
  n          = n,
  alpha_e    = alpha_e_s2,
  gamma_e    = gamma_e_s2,
  alpha_u    = alpha_u_s2,
  gamma_u    = gamma_u_s2,
  model_type = model_type
)

cat("Laplace MAP estimates:\n")
```

```
## Laplace MAP estimates:
```

```r
cat("beta:", laplace_s2$theta_map[1:p], "\n")
```

```
## beta: 4.90517 0.4732965 0.5168713 0.5460444
```

```r
cat("tau_e:", exp(laplace_s2$theta_map[p + q_s2 + 1]), "\n")
```

```
## tau_e: 5.199454
```

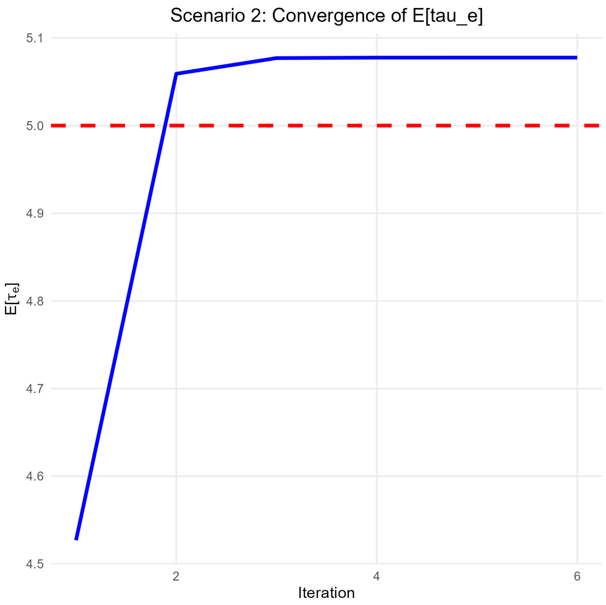
```r
cat("tau_u:", exp(laplace_s2$theta_map[p + q_s2 + 2]), "\n")
```

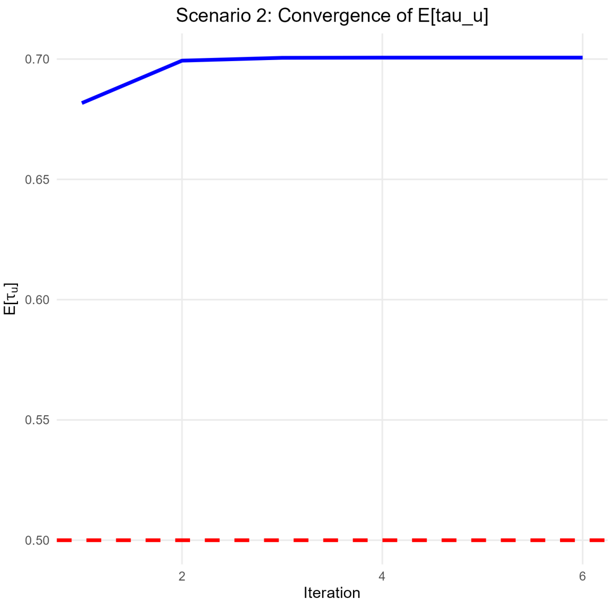
```
## tau_u: 0.6544494
```

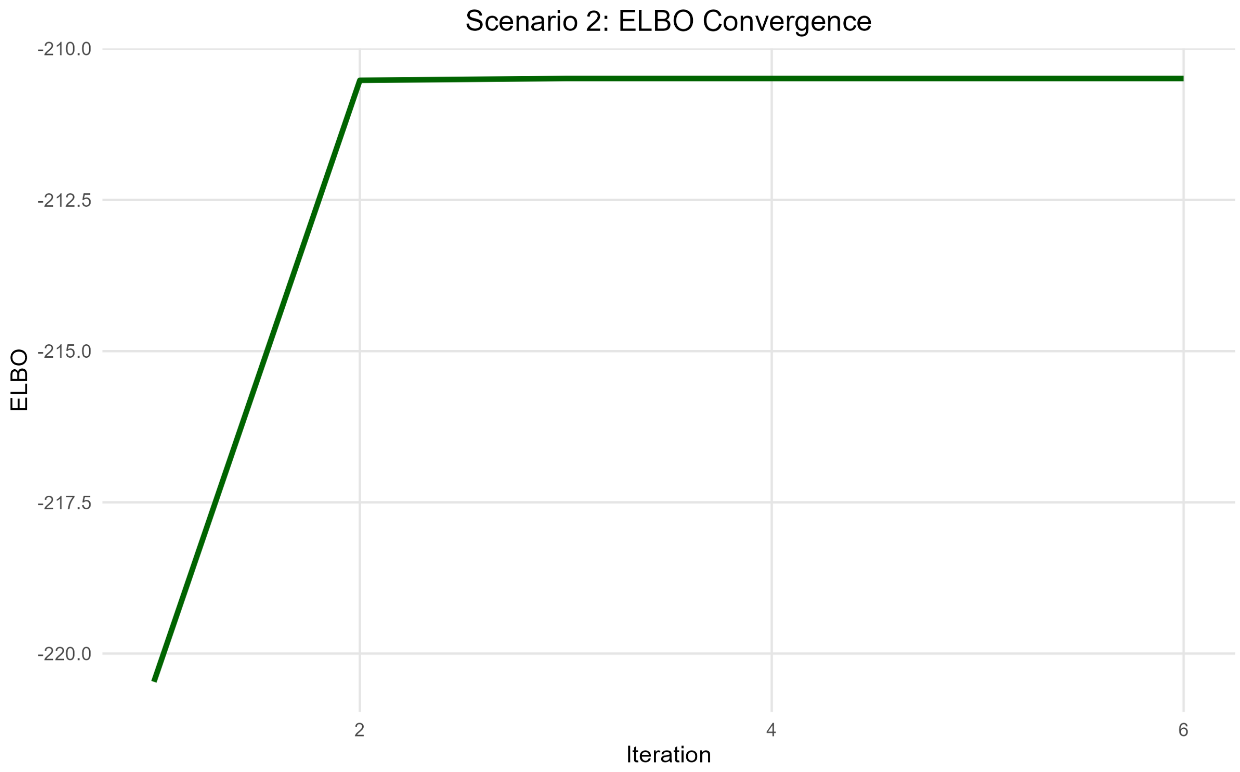
```r
# Save results for VI-COMPARE
saveRDS(list(
  vb = results_s2,
  laplace = laplace_s2,
  data = list(X = X_s2, Z = Z_s2, y = y_s2),
  truth = list(tau_u = tau_u_true, tau_e = tau_e_true),
  config = list(q = q_s2, nq = nq_6levels,
                alpha_u = alpha_u_s2, gamma_u = gamma_u_s2,
                prior_type = "strong")
), file.path("..", "results", "debug3_s2_strong.rds"))
```

```r
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_s2,
    gamma_e    = gamma_e_s2,
    alpha_u    = alpha_u_s2,
    gamma_u    = gamma_u_s2,
    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
}
```

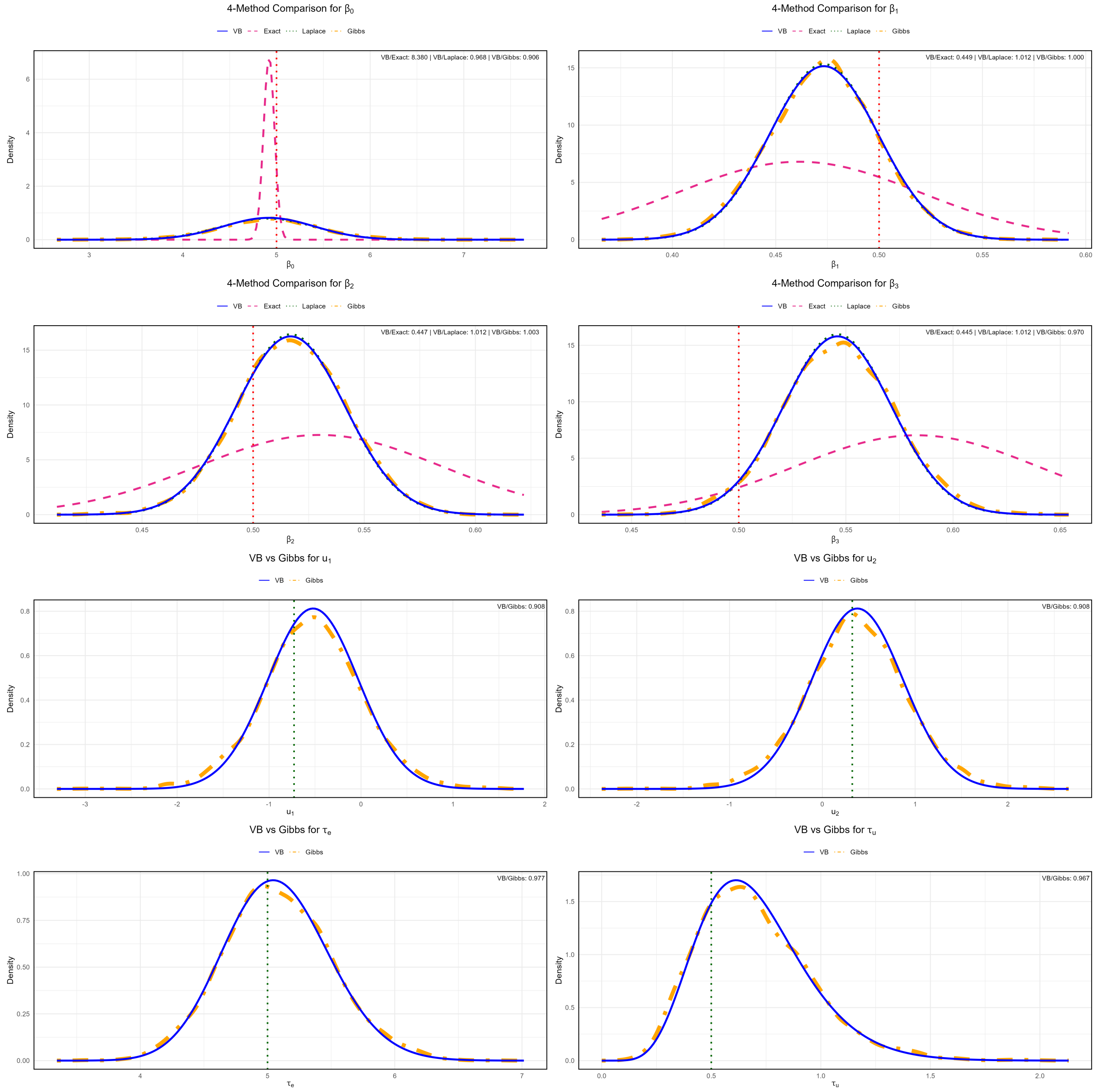
```
##
## Running Gibbs sampler...
## Gibbs posterior means:
## beta: 4.929035 0.4727112 0.5171898 0.546751
## tau_e: 5.081802
## tau_u: 0.6984903
```







```
## 6 levels: 8-panel ggplot saved
```


4-Method Comparison for β0
VB Exact Laplace Gibbs
VB/Exact: 8.380 | VB/Laplace: 0.968 | VB/Gibbs: 0.906
Density
0
2
4
6
3
4
5
6
7
β0
4-Method Comparison for β1
VB Exact Laplace Gibbs
VB/Exact: 0.449 | VB/Laplace: 1.012 | VB/Gibbs: 1.000
Density
0
5
10
15
0.40
0.45
0.50
0.55
0.60
β1
4-Method Comparison for β2
VB Exact Laplace Gibbs
VB/Exact: 0.447 | VB/Laplace: 1.012 | VB/Gibbs: 1.003
Density
0
5
10
15
0.45
0.50
0.55
0.60
β2
4-Method Comparison for β3
VB Exact Laplace Gibbs
VB/Exact: 0.445 | VB/Laplace: 1.012 | VB/Gibbs: 0.970
Density
0
5
10
15
0.45
0.50
0.55
0.60
0.65
β3
VB vs Gibbs for u1
VB Gibbs
VB/Gibbs: 0.908
Density
0.0
0.2
0.4
0.6
0.8
-3
-2
-1
0
1
2
u1
VB vs Gibbs for u2
VB Gibbs
VB/Gibbs: 0.908
Density
0.0
0.2
0.4
0.6
0.8
-2
-1
0
1
2
u2
VB vs Gibbs for τe
VB Gibbs
VB/Gibbs: 0.977
Density
0.00
0.25
0.50
0.75
1.00
4
5
6
7
τe
VB vs Gibbs for τu
VB Gibbs
VB/Gibbs: 0.967
Density
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
τu

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   4.9703154  5.0775615
## 2  E[tau_u]        0.5   0.9338213  0.7005995
## 3 sigma^2_e        0.2   0.2011945  0.1969449
## 4 sigma^2_u        2.0   1.0708687  1.4273490
```

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

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

```
cat("30 levels: 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")
```

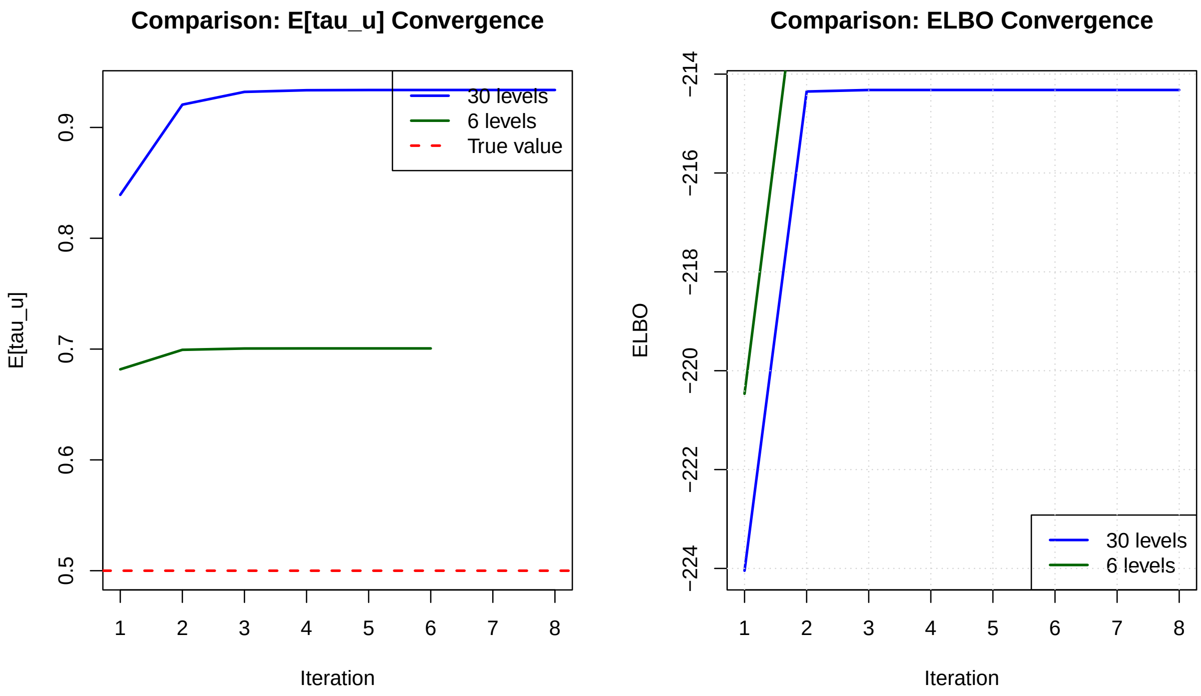
```
## 30 levels: VB tau_u = 0.9338 vs True = 0.5 (ratio: 1.8676 )
```

```
cat("6 levels:  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")
```

```
## 6 levels:  VB tau_u = 0.7006 vs True = 0.5 (ratio: 1.4012 )
```

```
par(mfrow = c(1, 2))

max_len <- max(length(results_s1$E_tau_u_history),
               length(results_s2$E_tau_u_history))

plot(
  1:length(results_s1$E_tau_u_history),
  results_s1$E_tau_u_history,
  type = 'l',
  lwd  = 2,
  col  = 'blue',
  xlab = 'Iteration',
  ylab = 'E[tau_u]',
  main = 'Comparison: E[tau_u] Convergence',
  xlim = c(1, max_len),
  ylim = range(c(results_s1$E_tau_u_history, results_s2$E_tau_u_history, tau_u_true)))
lines(1:length(results_s2$E_tau_u_history),
      results_s2$E_tau_u_history,
      col = 'darkgreen',
      lwd = 2)
```

```
abline(h = tau_u_true, col = 'red', lty = 2, lwd = 2)
legend('topright',
       legend = c('30 levels', '6 levels', 'True value'),
       col    = c('blue', 'darkgreen', 'red'),
       lty    = c(1, 1, 2),
       lwd    = 2)

plot(
  1:length(results_s1$elbo_history),
  results_s1$elbo_history,
  type = 'l',
  lwd  = 2,
  col  = 'blue',
  xlab = 'Iteration',
  ylab = 'ELBO',
  main = 'Comparison: ELBO Convergence',
  xlim = c(1, max_len))
lines(1:length(results_s2$elbo_history),
      results_s2$elbo_history,
      col = 'darkgreen',
      lwd = 2)
legend('bottomright',
       legend = c('30 levels', '6 levels'),
       col    = c('blue', 'darkgreen'),
       lty    = 1,
       lwd    = 2)
grid()
```


Comparison: E[tau_u] Convergence
E[tau_u]
Iteration
30 levels
6 levels
True value
Comparison: ELBO Convergence
ELBO
Iteration
30 levels
6 levels