STAT 447 Project:

Evaluating SNIS and MCMC for Bayesian Inference in High-Dimensional Biomedical Data

Yifei Li 25990359

Introduction:

In biomedical data analysis, accurate modeling and uncertainty quantification of physiological indicators are essential for reliable decision-making. The glycosylated hemoglobin level (glyhb) is a long-term measure of blood sugar and serves as a key biomarker in diabetes diagnosis and monitoring. Understanding how biological and demographic factors — such as age, weight, blood pressure, and cholesterol — relate to glyhb levels is crucial for clinical interpretation and intervention design. This project investigates Bayesian regression approaches for modeling glyhb as a function of physiological predictors. I focus on comparing two Bayesian estimator under increasing dimensional model: Markov Chain Monte Carlo and Sequential Importance Sampling. As discussed in the course, these methods differ substantially in their computational behavior and sampling efficiency. I use effective sample size and 95% credible interval width as key evaluation metrics. This work contributes to the theme of scientific comparison of Bayesian estimators under different model structures.

Dataset and Model Specification:

The dataset used in this project originates from the Vanderbilt Biostatistics Datasets, a public repository maintained by the Department of Biostatistics at Vanderbilt University. It contains biomedical records of 403 adult patients, which including various physiological measurements. The dataset contains 18 variables and glycosylated hemoglobin (glyhb) as the response variable (see figure 1 in Appendix). Among these, we select three different subsets of predictors to build models of increasing dimensionality.

- The low-dimensional model includes 5 core predictors only. They are age, weight, waist, bp.1d and bp.1s (first diastolic / systolic blood pressure).
- The mid-dimensional model expands to 9 variables by adding chol (cholesterol), stab.glu (stabilized glucose), hdl (HDL cholesterol), and height.
- The high-dimensional model keeps all 9 variables in mid-dimensional mode then further introduces a hierarchical model, allowing intercept, weight, and waist to change by group. Here, use patients frame as group variable (small/medium/large).

For the math formula of 3 models, check figure 2,3 and 4.

Method Implementation:

I use Bayesian regression models across three levels of dimensionality using MCMC and SNIS. This section outlines how both approaches are implemented. All predictors are standardized before modeling.

Markov Chain Monte Carlo (MCMC):

For each model (low, mid, and high dimension), we perform posterior by HMC in Stan. Each model is assigned weakly informative priors. Each MCMC runs 4 chains, each with 2000 iterations that including 1000 warm up steps. **(See figure 5, 6 and 7)**

Sequential Importance Sampling (SNIS):

To approximate the same posterior distributions via importance sampling, I apply SNIS with a full posterior proposal. Here, Use MCMC samples to estimate the posterior mean and covariance to construct a normal proposal from this posterior approximation. SNIS Draw 10,000 samples from the proposal and compute weights using likelihood terms (focuses purely on likelihood efficiency). Because SNIS performance is highly sensitive to the quality of the proposal distribution, we intentionally choose a strong proposal based on MCMC output. This project is about whether SNIS can remain competitive as model dimensionality increases. Using a high-quality proposal makes the comparison more interpretable since it examines its upper-bound performance. **(See figure 8, 9 and 10)**

Evaluation Metrics:

Here, the metrics that I choose are: Effective Sample Size and 95% Credible Interval.

Effective Sample Size (ESS):

For MCMC, Stan computes ESS by accounting for autocorrelation within Markov chains, reflecting how well the posterior is explored.

For SNIS, we use the standard importance sampling ESS formula: $\mathrm{ESS} = \frac{1}{\sum w_i^2}$

95% Credible Interval Width:

For MCMC, use Stan's built-in quantile function across posterior draws.

For SNIS, since each sample has a weight, so we can sort the samples, then normalized weights to find where cumulative weight reaches 2.5% and 97.5%

ESS measures how efficiently we sample the posterior and CI width captures the precision of the estimates. narrow CI implies higher estimation precision.

Analysis:

We evaluate the posterior estimate of MCMC and SNIS using ESS and 95% CI. The comparison is performed separately across low, mid, and high-dimensional models. In the low-dimensional setting, SNIS achieves higher ESS percentages than MCMC across all parameters. The credible intervals from SNIS closely match those from

MCMC, suggesting comparable posterior estimation accuracy. **(See figure 11 and 14)**

In the mid-dimensional model, both methods perform similarly well. SNIS maintains near-maximum ESS for most parameters, and CI widths remain nearly identical to those from MCMC, indicating that the posterior approximation remains reliable as dimensionality increases. **(See figure 12 and 15)**

However, in the high-dimensional model, SNIS performance declines substantially. ESS values drop below 60% for most parameters and some SNIS credible intervals become wider than MCMC's. This indicate that SNIS performs poorly, even when the proposal distribution is informed by the full posterior. **(See figure 13 and 16)**

Discussion:

SNIS's performance will decrease a lot in high-dimensional spaces. As Chatterjee & Diaconis (2018) showed that the number of samples required for SNIS to grows exponentially when dimension goes up unless the proposal distribution is perfectly same with the target. This limitation proved in this project too.

To mitigate such issues, Agapiou et al. (2017) provide one approach to analyze the intrinsic dimension of the problem, which more accurately predicts the sample size and cost of importance sampling than the original distribution dimension.

Conclusion:

We compare MCMC and SNIS for Bayesian regression across models of increasing dimensionality. SNIS performs well in low and mid dimensions but drops significantly in high-dimensional settings. In contrast, MCMC maintains stable performance across all dimensions. It makes MCMC be a better Bayesian estimator for complex models.

References

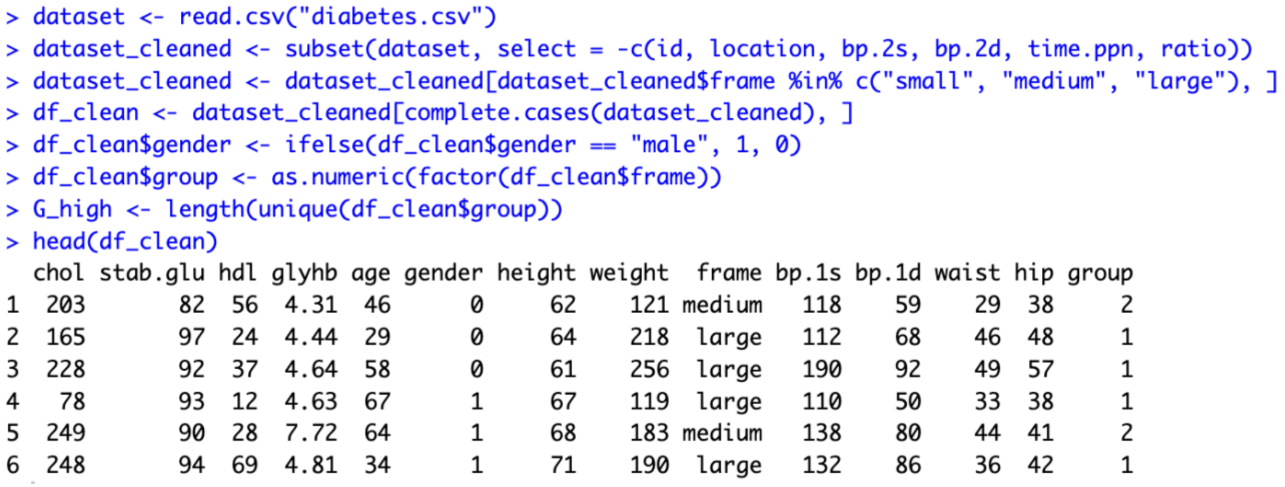
1. Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., & Stuart, A. M. (2017). *Importance sampling: Intrinsic dimension and computational cost*. arXiv preprint arXiv:1511.06196

2. Chatterjee, S., & Diaconis, P. (2018). The sample size required in importance sampling. *Annals of Applied Probability*, 28(2), 1099–1135. https://doi.org/10.1214/17-AAP1326

Appendix:

Path for github: https://github.com/YifeiLi0/STAT447_final_project

Figure 1: first few lines of clean data



```
> dataset <- read.csv("diabetes.csv")
> dataset_cleaned <- subset(dataset, select = -c(id, location, bp.2s, bp.2d, time.ppn, ratio))
> dataset_cleaned <- dataset_cleaned[dataset_cleaned$frame %in% c("small", "medium", "large"), ]
> df_clean <- dataset_cleaned[complete.cases(dataset_cleaned), ]
> df_clean$gender <- ifelse(df_clean$gender == "male", 1, 0)
> df_clean$group <- as.numeric(factor(df_clean$frame))
> G_high <- length(unique(df_clean$group))
> head(df_clean)
  chol stab.glu hdl glyhb age gender height weight  frame bp.1s bp.1d waist hip group
1  203       82  56  4.31  46      0     62    121 medium   118    59    29  38     2
2  165       97  24  4.44  29      0     64    218  large   112    68    46  48     1
3  228       92  37  4.64  58      0     61    256  large   190    92    49  57     1
4   78       93  12  4.63  67      1     67    119  large   110    50    33  38     1
5  249       90  28  7.72  64      1     68    183 medium   138    80    44  41     2
6  248       94  69  4.81  34      1     71    190  large   132    86    36  42     1
```

Figure 2: Low-Dimensional model

$$
\begin{aligned}
y_i &\sim \mathcal{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{weight}} \cdot \text{weight}_i + \beta_{\text{waist}} \cdot \text{waist}_i \\
&\quad + \beta_{\text{bp1d}} \cdot \text{bp1d}_i + \beta_{\text{bp1s}} \cdot \text{bp1s}_i \\
\beta_j &\sim \mathcal{N}(0, 2^2), \quad \sigma \sim \text{Cauchy}^+(0, 2.5)
\end{aligned}
$$

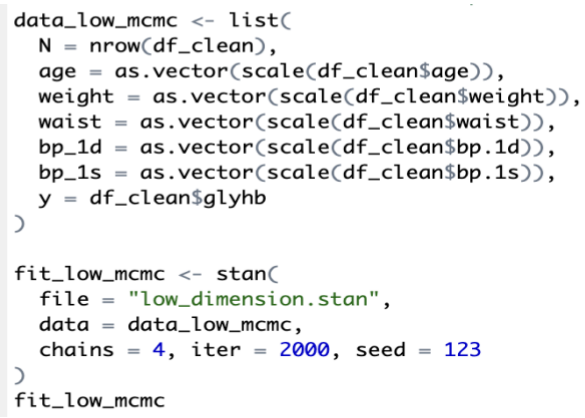
Figure 3: Mid-Dimensional model.

$$
\begin{aligned}
y_i &\sim \mathcal{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{weight}} \cdot \text{weight}_i + \beta_{\text{waist}} \cdot \text{waist}_i \\
&\quad + \beta_{\text{bp1d}} \cdot \text{bp1d}_i + \beta_{\text{bp1s}} \cdot \text{bp1s}_i + \beta_{\text{chol}} \cdot \text{chol}_i \\
&\quad + \beta_{\text{stabglu}} \cdot \text{stabglu}_i + \beta_{\text{hdl}} \cdot \text{hdl}_i + \beta_{\text{height}} \cdot \text{height}_i \\
\beta_j &\sim \mathcal{N}(0, 2^2), \quad \sigma \sim \text{Cauchy}^+(0, 2.5)
\end{aligned}
$$

Figure 4: High-Dimensional model (Hierarchical on Intercept, Weight, Waist)

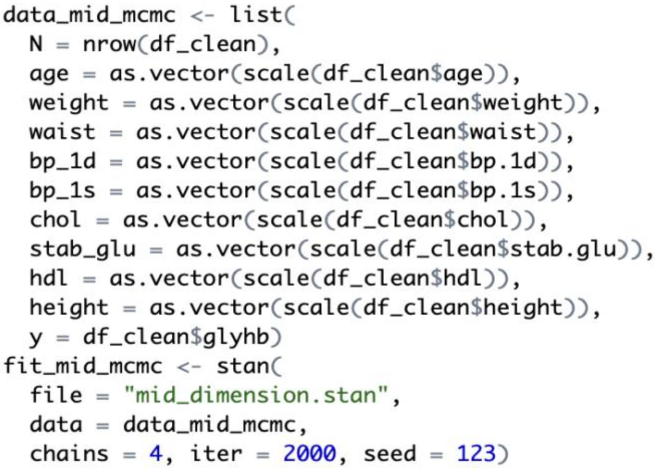
$$
\begin{aligned}
y_i &\sim \mathcal{N}(\mu_i, \sigma^2) \\
\mu_i &= \alpha_{g[i]} + \beta_{\text{weight},g[i]} \cdot \text{weight}_i + \beta_{\text{waist},g[i]} \cdot \text{waist}_i \\
&\quad + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{bp1d}} \cdot \text{bp1d}_i + \beta_{\text{bp1s}} \cdot \text{bp1s}_i \\
&\quad + \beta_{\text{chol}} \cdot \text{chol}_i + \beta_{\text{stabglu}} \cdot \text{stabglu}_i + \beta_{\text{hdl}} \cdot \text{hdl}_i + \beta_{\text{height}} \cdot \text{height}_i \\
\alpha_g &\sim \mathcal{N}(0, 5^2), \quad \beta_{\text{weight},g}, \beta_{\text{waist},g} \sim \mathcal{N}(0, 2^2) \\
\beta_j &\sim \mathcal{N}(0, 2^2), \quad \sigma \sim \text{Cauchy}^+(0, 2.5)
\end{aligned}
$$

Figure 5: Low-dimensional model code for MCMC



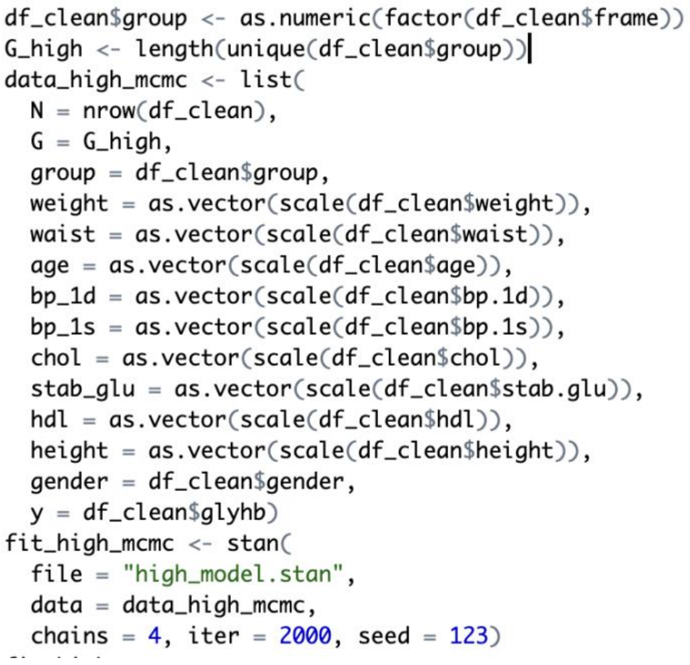
```
data_low_mcmc <- list(
  N = nrow(df_clean),
  age = as.vector(scale(df_clean$age)),
  weight = as.vector(scale(df_clean$weight)),
  waist = as.vector(scale(df_clean$waist)),
  bp_1d = as.vector(scale(df_clean$bp.1d)),
  bp_1s = as.vector(scale(df_clean$bp.1s)),
  y = df_clean$glyhb
)

fit_low_mcmc <- stan(
  file = "low_dimension.stan",
  data = data_low_mcmc,
  chains = 4, iter = 2000, seed = 123
)
fit_low_mcmc
```

Figure 6: Mid-dimensional model code for MCMC



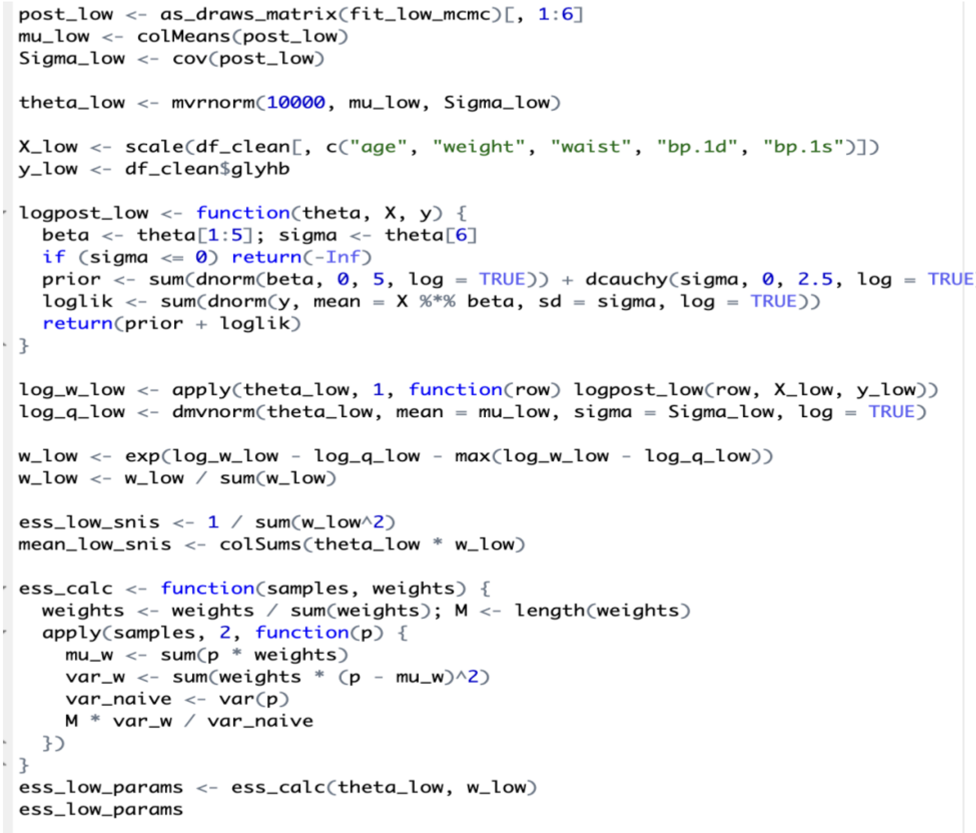
```
data_mid_mcmc <- list(
  N = nrow(df_clean),
  age = as.vector(scale(df_clean$age)),
  weight = as.vector(scale(df_clean$weight)),
  waist = as.vector(scale(df_clean$waist)),
  bp_1d = as.vector(scale(df_clean$bp.1d)),
  bp_1s = as.vector(scale(df_clean$bp.1s)),
  chol = as.vector(scale(df_clean$chol)),
  stab_glu = as.vector(scale(df_clean$stab.glu)),
  hdl = as.vector(scale(df_clean$hdl)),
  height = as.vector(scale(df_clean$height)),
  y = df_clean$glyhb)
fit_mid_mcmc <- stan(
  file = "mid_dimension.stan",
  data = data_mid_mcmc,
  chains = 4, iter = 2000, seed = 123)
```

Figure 7: High-dimensional model code for MCMC



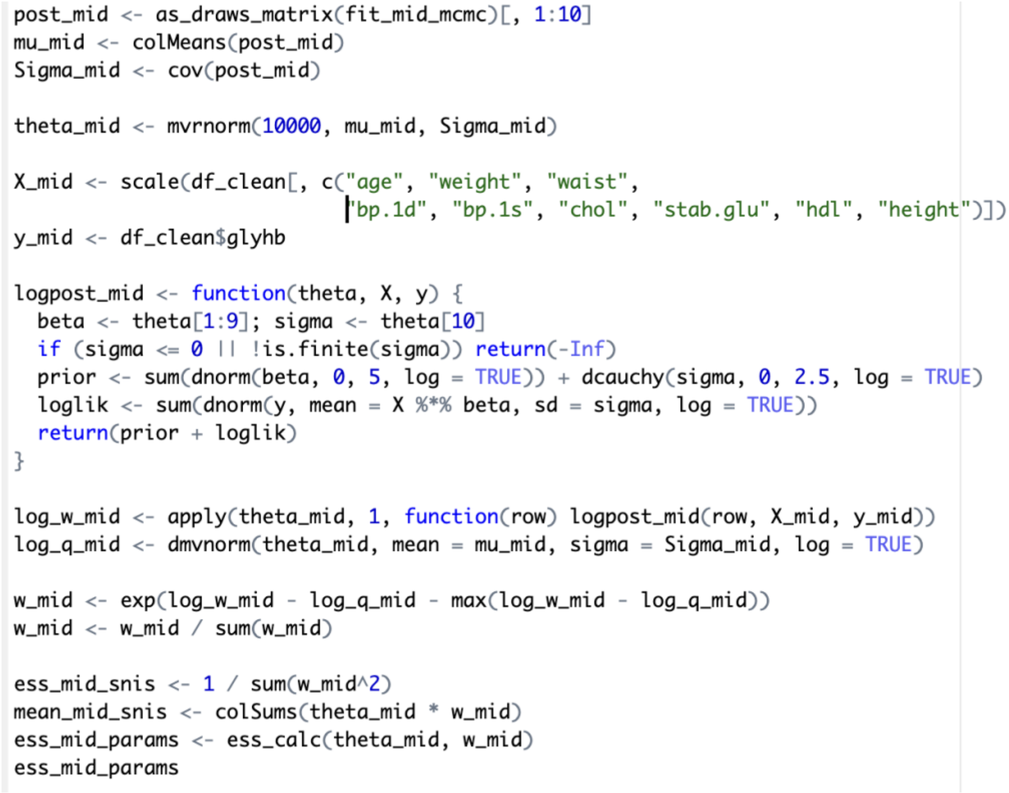
```
df_clean$group <- as.numeric(factor(df_clean$frame))
G_high <- length(unique(df_clean$group))
data_high_mcmc <- list(
  N = nrow(df_clean),
  G = G_high,
  group = df_clean$group,
  weight = as.vector(scale(df_clean$weight)),
  waist = as.vector(scale(df_clean$waist)),
  age = as.vector(scale(df_clean$age)),
  bp_1d = as.vector(scale(df_clean$bp.1d)),
  bp_1s = as.vector(scale(df_clean$bp.1s)),
  chol = as.vector(scale(df_clean$chol)),
  stab_glu = as.vector(scale(df_clean$stab.glu)),
  hdl = as.vector(scale(df_clean$hdl)),
  height = as.vector(scale(df_clean$height)),
  gender = df_clean$gender,
  y = df_clean$glyhb)
fit_high_mcmc <- stan(
  file = "high_model.stan",
  data = data_high_mcmc,
  chains = 4, iter = 2000, seed = 123)
```

Figure 8: Low-dimensional model code for SNIS



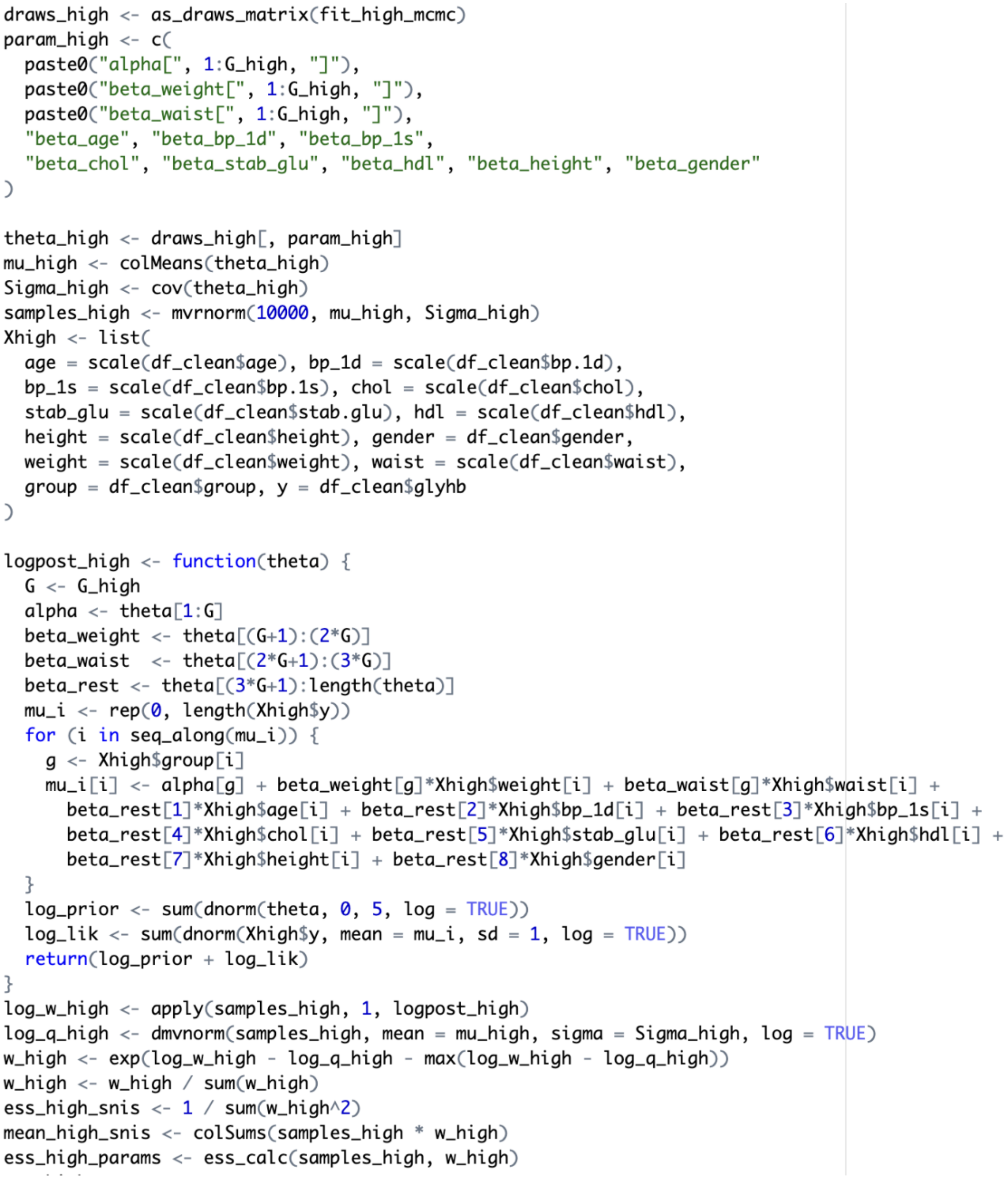
```
post_low <- as_draws_matrix(fit_low_mcmc)[, 1:6]
mu_low <- colMeans(post_low)
Sigma_low <- cov(post_low)

theta_low <- mvrnorm(10000, mu_low, Sigma_low)

X_low <- scale(df_clean[, c("age", "weight", "waist", "bp.1d", "bp.1s")])
y_low <- df_clean$glyhb

logpost_low <- function(theta, X, y) {
  beta <- theta[1:5]; sigma <- theta[6]
  if (sigma <= 0) return(-Inf)
  prior <- sum(dnorm(beta, 0, 5, log = TRUE)) + dcauchy(sigma, 0, 2.5, log = TRUE)
  loglik <- sum(dnorm(y, mean = X %*% beta, sd = sigma, log = TRUE))
  return(prior + loglik)
}

log_w_low <- apply(theta_low, 1, function(row) logpost_low(row, X_low, y_low))
log_q_low <- dmvnorm(theta_low, mean = mu_low, sigma = Sigma_low, log = TRUE)

w_low <- exp(log_w_low - log_q_low - max(log_w_low - log_q_low))
w_low <- w_low / sum(w_low)

ess_low_snis <- 1 / sum(w_low^2)
mean_low_snis <- colSums(theta_low * w_low)

ess_calc <- function(samples, weights) {
  weights <- weights / sum(weights); M <- length(weights)
  apply(samples, 2, function(p) {
    mu_w <- sum(p * weights)
    var_w <- sum(weights * (p - mu_w)^2)
    var_naive <- var(p)
    M * var_w / var_naive
  })
}
ess_low_params <- ess_calc(theta_low, w_low)
ess_low_params
```

Figure 9: mid-dimensional model code for SNIS



```
post_mid <- as_draws_matrix(fit_mid_mcmc)[, 1:10]
mu_mid <- colMeans(post_mid)
Sigma_mid <- cov(post_mid)

theta_mid <- mvrnorm(10000, mu_mid, Sigma_mid)

X_mid <- scale(df_clean[, c("age", "weight", "waist",
                            "bp.1d", "bp.1s", "chol", "stab.glu", "hdl", "height")])
y_mid <- df_clean$glyhb

logpost_mid <- function(theta, X, y) {
  beta <- theta[1:9]; sigma <- theta[10]
  if (sigma <= 0 || !is.finite(sigma)) return(-Inf)
  prior <- sum(dnorm(beta, 0, 5, log = TRUE)) + dcauchy(sigma, 0, 2.5, log = TRUE)
  loglik <- sum(dnorm(y, mean = X %*% beta, sd = sigma, log = TRUE))
  return(prior + loglik)
}

log_w_mid <- apply(theta_mid, 1, function(row) logpost_mid(row, X_mid, y_mid))
log_q_mid <- dmvnorm(theta_mid, mean = mu_mid, sigma = Sigma_mid, log = TRUE)

w_mid <- exp(log_w_mid - log_q_mid - max(log_w_mid - log_q_mid))
w_mid <- w_mid / sum(w_mid)

ess_mid_snis <- 1 / sum(w_mid^2)
mean_mid_snis <- colSums(theta_mid * w_mid)
ess_mid_params <- ess_calc(theta_mid, w_mid)
ess_mid_params
```

Figure 10: High-dimensional model code for SNIS



```
draws_high <- as_draws_matrix(fit_high_mcmc)
param_high <- c(
  paste0("alpha[", 1:G_high, "]"),
  paste0("beta_weight[", 1:G_high, "]"),
  paste0("beta_waist[", 1:G_high, "]"),
  "beta_age", "beta_bp_1d", "beta_bp_1s",
  "beta_chol", "beta_stab_glu", "beta_hdl", "beta_height", "beta_gender"
)

theta_high <- draws_high[, param_high]
mu_high <- colMeans(theta_high)
Sigma_high <- cov(theta_high)
samples_high <- mvrnorm(10000, mu_high, Sigma_high)
Xhigh <- list(
  age = scale(df_clean$age), bp_1d = scale(df_clean$bp.1d),
  bp_1s = scale(df_clean$bp.1s), chol = scale(df_clean$chol),
  stab_glu = scale(df_clean$stab.glu), hdl = scale(df_clean$hdl),
  height = scale(df_clean$height), gender = df_clean$gender,
  weight = scale(df_clean$weight), waist = scale(df_clean$waist),
  group = df_clean$group, y = df_clean$glyhb
)

logpost_high <- function(theta) {
  G <- G_high
  alpha <- theta[1:G]
  beta_weight <- theta[(G+1):(2*G)]
  beta_waist  <- theta[(2*G+1):(3*G)]
  beta_rest <- theta[(3*G+1):length(theta)]
  mu_i <- rep(0, length(Xhigh$y))
  for (i in seq_along(mu_i)) {
    g <- Xhigh$group[i]
    mu_i[i] <- alpha[g] + beta_weight[g]*Xhigh$weight[i] + beta_waist[g]*Xhigh$waist[i] +
      beta_rest[1]*Xhigh$age[i] + beta_rest[2]*Xhigh$bp_1d[i] + beta_rest[3]*Xhigh$bp_1s[i] +
      beta_rest[4]*Xhigh$chol[i] + beta_rest[5]*Xhigh$stab_glu[i] + beta_rest[6]*Xhigh$hdl[i] +
      beta_rest[7]*Xhigh$height[i] + beta_rest[8]*Xhigh$gender[i]
  }
  log_prior <- sum(dnorm(theta, 0, 5, log = TRUE))
  log_lik <- sum(dnorm(Xhigh$y, mean = mu_i, sd = 1, log = TRUE))
  return(log_prior + log_lik)
}
log_w_high <- apply(samples_high, 1, logpost_high)
log_q_high <- dmvnorm(samples_high, mean = mu_high, sigma = Sigma_high, log = TRUE)
w_high <- exp(log_w_high - log_q_high - max(log_w_high - log_q_high))
w_high <- w_high / sum(w_high)
ess_high_snis <- 1 / sum(w_high^2)
mean_high_snis <- colSums(samples_high * w_high)
ess_high_params <- ess_calc(samples_high, w_high)
```

Figure 11:

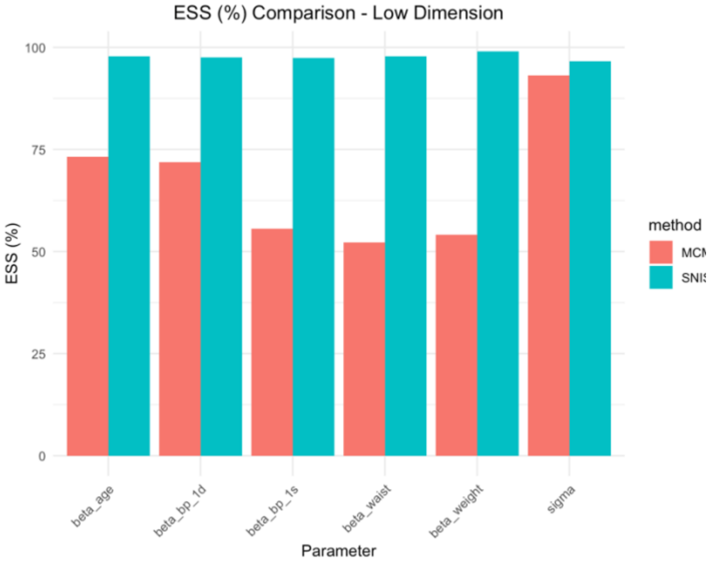


Figure 12:

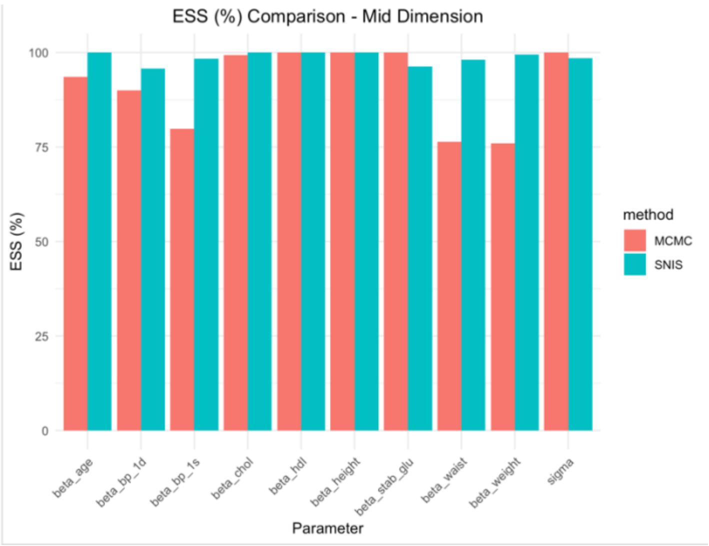


Figure 13:

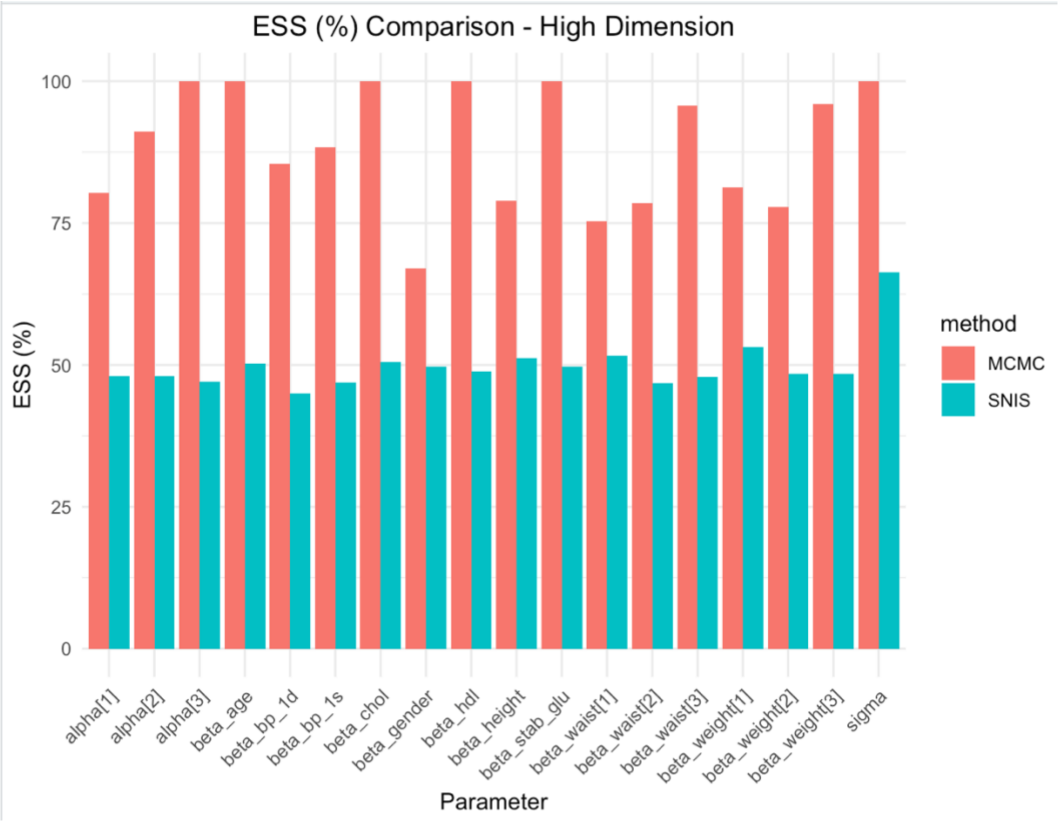


Figure 14:

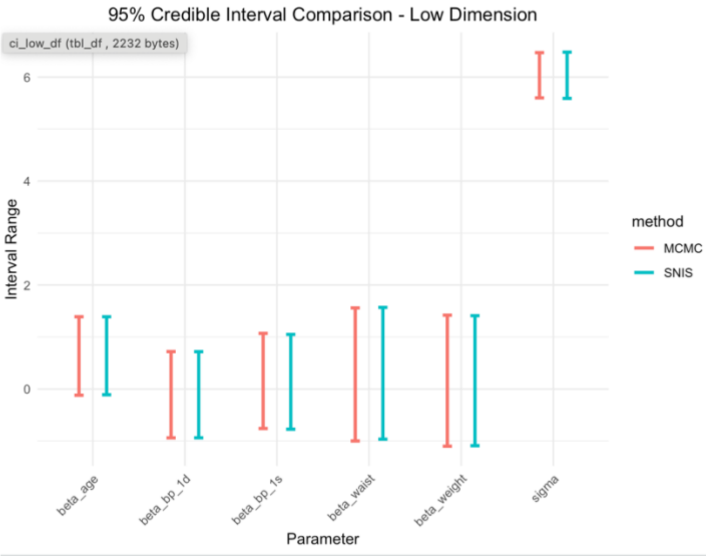


Figure 15:

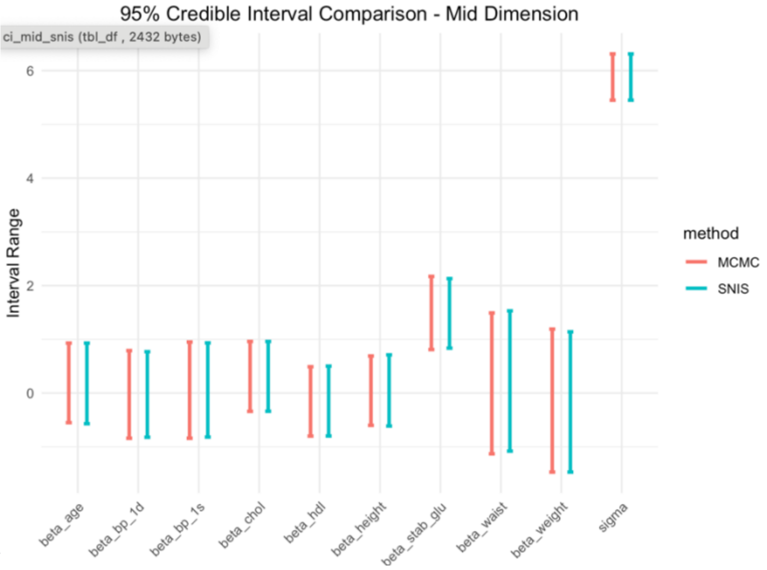
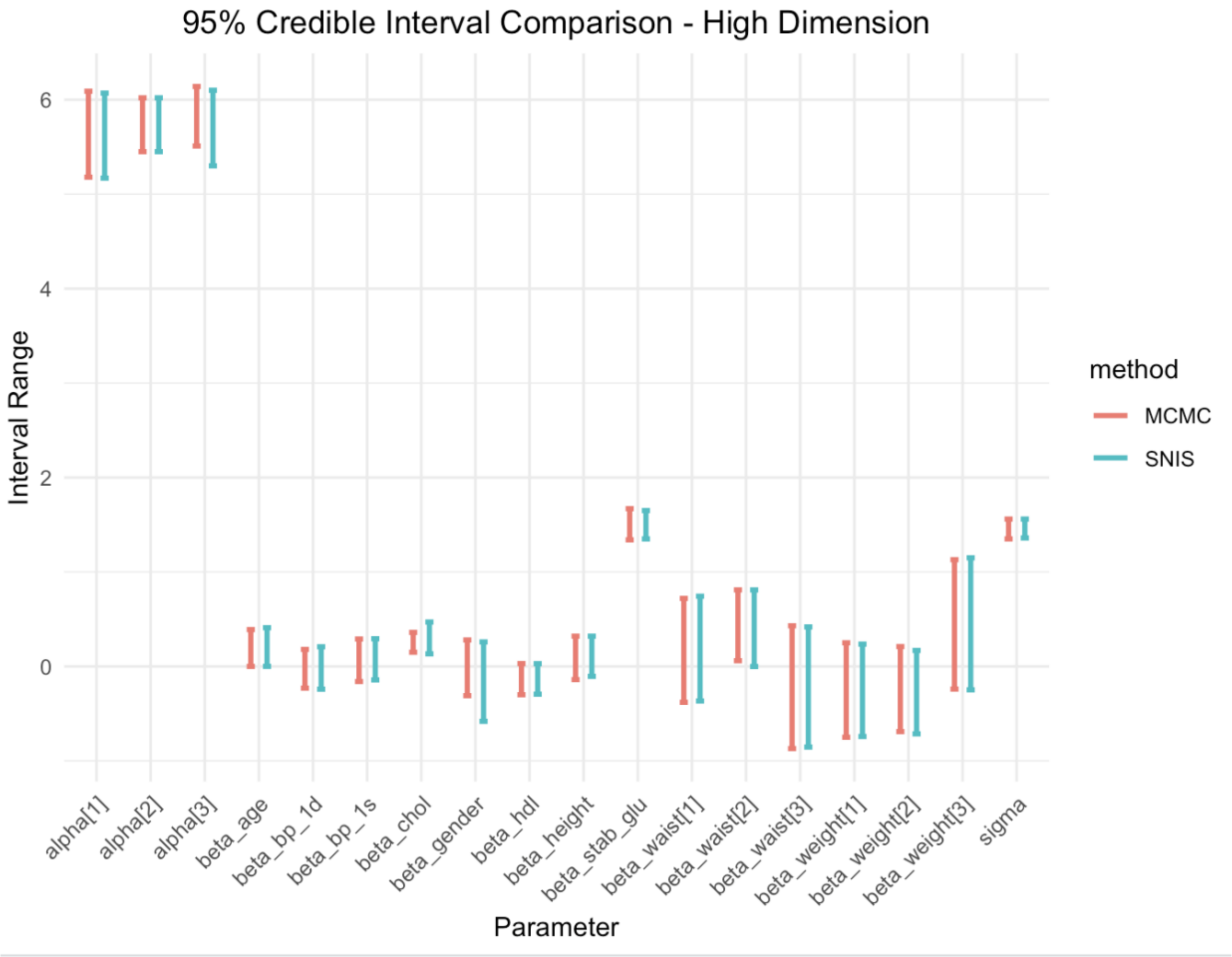


Figure 16:


95% Credible Interval Comparison - High Dimension
Interval Range
alpha[1]
alpha[2]
alpha[3]
beta_age
beta_bp_1d
beta_bp_1s
beta_chol
beta_gender
beta_hdl
beta_height
beta_stab_glu
beta_waist[1]
beta_waist[2]
beta_waist[3]
beta_weight[1]
beta_weight[2]
beta_weight[3]
sigma
Parameter
method
MCMC
SNIS