Bayesian Logistic Regression for Heart Disease: Model Comparison and Predictive-Prescriptive Analysis

Min Shun

Bina Nusantara University

October 2025

1. Introduction

This project analyzes a **heart disease dataset** collected in 1988 from four sources: **Cleveland, Hungary, Switzerland, and the Long Beach V databases**. The dataset has been **preprocessed and scaled** to standardize covariates and facilitate model convergence. The primary goal is to explore **Bayesian logistic regression** for modeling the presence of heart disease, while systematically comparing the performance of two approaches:

1. **Baseline Model** → specified with **uninformative priors.**
2. **Hypertuned Model** → specified with **informative priors** derived from previous research to incorporate domain knowledge.

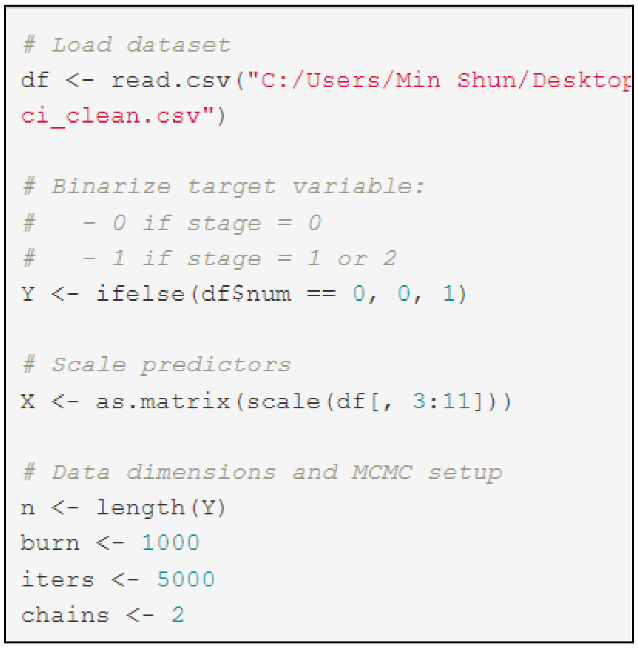
The **target variable is binary**, indicating whether a patient has **no heart disease (stage 0)** or **some degree of heart disease (stage 1 or 2)**. The predictors include a set of **standardized** clinical features such as age, sex, resting blood pressure, cholesterol, fasting blood sugar, maximum heart rate, exercise-induced angina, ST-segment depression, and slope of the ST segment during exercise.

The objectives of this report are fourfold:

- To evaluate **convergence diagnostics** for both Bayesian models.
- To **compare models** using Deviance Information Criterion (DIC) and Watanabe-Akaike Information Criterion (WAIC).
- To **identify clinical covariates** with significant **posterior effects** on heart disease.
- To perform **predictive and prescriptive decision-making using Thompson Sampling** on the best performing model, demonstrating how **posterior uncertainty** can inform individualized treatment decisions and **minimize expected clinical cost.**

By combining **model comparison** with **decision analysis**, this project provides not only **statistical insights** into the relative performance of Bayesian priors but also a practical demonstration of how **Bayesian inference** can guide **clinical decision-making under uncertainty.**

2. Variable Declaration and Data Preparation



```
# Load dataset
df <- read.csv("C:/Users/Min Shun/Desktop
ci_clean.csv")

# Binarize target variable:
#   - 0 if stage = 0
#   - 1 if stage = 1 or 2
Y <- ifelse(df$num == 0, 0, 1)

# Scale predictors
X <- as.matrix(scale(df[, 3:11]))

# Data dimensions and MCMC setup
n <- length(Y)
burn <- 1000
iters <- 5000
chains <- 2
```

The dataset is divided into two main components:

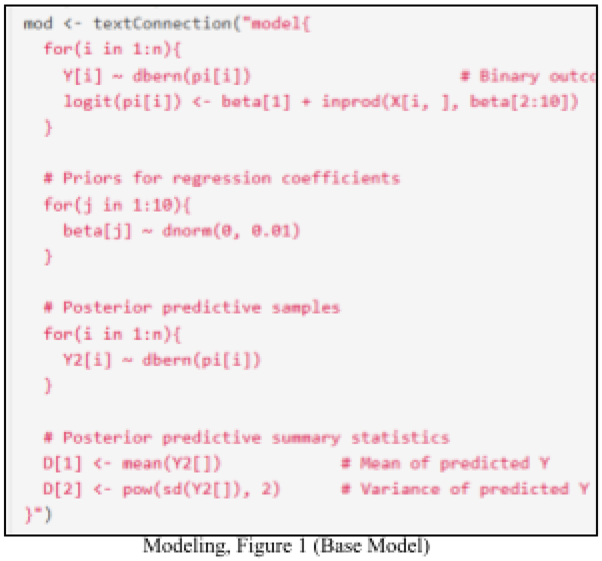
- **Target variable (Y):** A binary indicator of **heart disease presence.**
  - **0 = no heart disease (stage 0)**
  - **1 = presence of heart disease (stage 1 or 2).**
- **Predictors (X): Scaled covariates** consisting of the main **clinical features: *age, sex, trestbps, chol, fbs, thalach, exang, oldpeak,* and *slope*.** Although the raw data values are already standardized, **additional scaling is applied** to ensure that predictors are on a **comparable scale.** This adjustment improves the **effectiveness of informative priors** with small variance by preventing extreme values from dominating the posterior distribution. Consequently, **scaling** helps **stabilize parameter estimation** and facilitates more **efficient MCMC sampling.**

In addition, the **data dimensions and MCMC hyperparameters** are defined to guide the Bayesian analysis:

- **Chains (chains = 2):** Running multiple chains enables **convergence diagnostics**. If both chains converge to the same posterior distribution, it indicates that the **sampler** has adequately explored the parameter space. Two chains are considered a **minimal yet sufficient** choice for this purpose.

- **Burn-in (burn = 1000):** The initial samples in each chain are discarded to allow the **sampler** to reach the **stationary distribution** of the posterior, thereby reducing bias from arbitrary initial values.
- **Iterations (iters = 5000):** Specifies the number of **posterior samples** drawn per chain. A larger number of iterations provides more precise estimates but increases **computational cost.** Here, 5000 iterations are selected as a **balance between accuracy and efficiency.**

3. Modeling



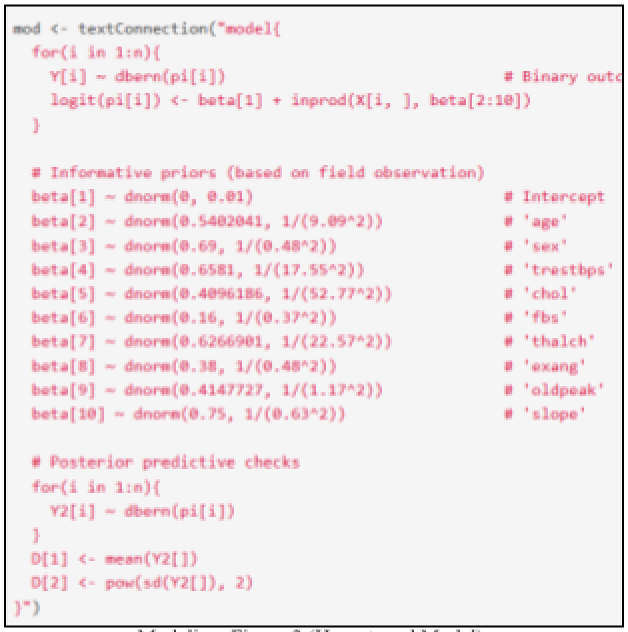
```
mod <- textConnection("model{
  for(i in 1:n){
    Y[i] ~ dbern(pi[i])                        # Binary outc
    logit(pi[i]) <- beta[1] + inprod(X[i, ], beta[2:10])
  }

  # Priors for regression coefficients
  for(j in 1:10){
    beta[j] ~ dnorm(0, 0.01)
  }

  # Posterior predictive samples
  for(i in 1:n){
    Y2[i] ~ dbern(pi[i])
  }

  # Posterior predictive summary statistics
  D[1] <- mean(Y2[])            # Mean of predicted Y
  D[2] <- pow(sd(Y2[]), 2)      # Variance of predicted Y
}")
```

Modeling, Figure 1 (Base Model)



```
mod <- textConnection("model{
  for(i in 1:n){
    Y[i] ~ dbern(pi[i])                                   # Binary outc
    logit(pi[i]) <- beta[1] + inprod(X[i, ], beta[2:10])
  }

  # Informative priors (based on field observation)
  beta[1] ~ dnorm(0, 0.01)                                # Intercept
  beta[2] ~ dnorm(0.5402041, 1/(9.09^2))                  # 'age'
  beta[3] ~ dnorm(0.69, 1/(0.48^2))                       # 'sex'
  beta[4] ~ dnorm(0.6581, 1/(17.55^2))                    # 'trestbps'
  beta[5] ~ dnorm(0.4096186, 1/(52.77^2))                 # 'chol'
  beta[6] ~ dnorm(0.16, 1/(0.37^2))                       # 'fbs'
  beta[7] ~ dnorm(0.6266901, 1/(22.57^2))                 # 'thalch'
  beta[8] ~ dnorm(0.38, 1/(0.48^2))                       # 'exang'
  beta[9] ~ dnorm(0.4147727, 1/(1.17^2))                  # 'oldpeak'
  beta[10] ~ dnorm(0.75, 1/(0.63^2))                      # 'slope'

  # Posterior predictive checks
  for(i in 1:n){
    Y2[i] ~ dbern(pi[i])
  }
  D[1] <- mean(Y2[])
  D[2] <- pow(sd(Y2[]), 2)
}")
```

Modeling, Figure 2 (Hypertuned Model)

Two **Bayesian logistic regression models** were specified using **JAGS**. Both models assume a **Bernoulli likelihood** for the **binary outcome variable Y**, with a **logit link function** to relate predictors to the probability **of heart disease**:

$$Y_i \sim \text{Bernoulli}(\pi_i), \quad \text{logit}(\pi_i) = \beta_1 + \sum_{j=2}^{10} X_{ij}\beta_j$$

where **β1 denotes the intercept**, and **βj are the regression coefficients** associated with the covariates.

The two models differ in the specification of their prior distributions:

1. **Uninformative Priors (Baseline Model)**

   In the initial model, **uninformative priors** were assigned to the **intercept** and **regression coefficients**, specified as **Normal distributions** with **mean 0 and variance 100**. This choice of prior reflects **minimal prior knowledge** and allows the data to primarily drive

posterior inference. This model serves as the **baseline Bayesian logistic regression model** (Figure 1).

2. **Informative Priors (Hypertuned Model)**

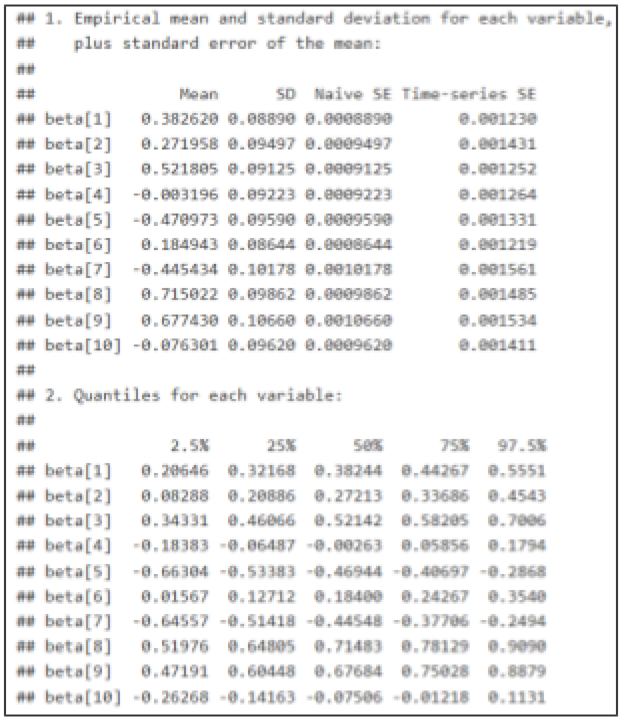
   In the second model, **informative priors** were incorporated using parameter estimates **derived from prior research by Ahamad et al. (2023).** For each covariate, **prior mean** and **variance value**s were extracted and subsequently scaled to align with the **standardized predictors** in this dataset. By doing so, the priors provide stronger guidance for **posterior estimation** while remaining consistent with the scale of the data. This specification represents a **hypertuned model,** reflecting the integration of external domain knowledge into the Bayesian framework (Figure 2).

4. Summary of Sample

Before comparing **model convergence**, it's useful to look at the **summary of the posterior samples (hypertuned vs base model)**:

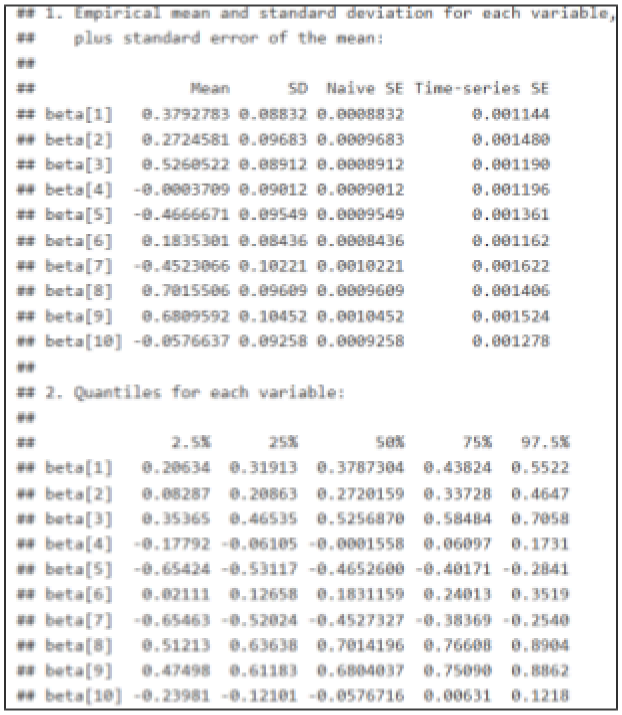
Sample Summary

- **Iterations:** 2001–7000
- **Thinning interval:** 1
- **Chains:** 2
- **Sample size per chain:** 5000



```
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## beta[1]   0.382620 0.08890 0.0008890       0.001230
## beta[2]   0.271958 0.09497 0.0009497       0.001431
## beta[3]   0.521805 0.09125 0.0009125       0.001252
## beta[4]  -0.003196 0.09223 0.0009223       0.001264
## beta[5]  -0.470973 0.09590 0.0009590       0.001331
## beta[6]   0.184943 0.08644 0.0008644       0.001219
## beta[7]  -0.445434 0.10178 0.0010178       0.001561
## beta[8]   0.715022 0.09862 0.0009862       0.001485
## beta[9]   0.677430 0.10660 0.0010660       0.001534
## beta[10] -0.076301 0.09620 0.0009620       0.001411
##
## 2. Quantiles for each variable:
##
##               2.5%      25%      50%      75%   97.5%
## beta[1]   0.20646  0.32168  0.38244  0.44267  0.5551
## beta[2]   0.08288  0.20886  0.27213  0.33686  0.4543
## beta[3]   0.34331  0.46066  0.52142  0.58205  0.7006
## beta[4]  -0.18383 -0.06487 -0.00263  0.05856  0.1794
## beta[5]  -0.66304 -0.53383 -0.46944 -0.40697 -0.2868
## beta[6]   0.01567  0.12712  0.18400  0.24267  0.3540
## beta[7]  -0.64557 -0.51418 -0.44548 -0.37706 -0.2494
## beta[8]   0.51976  0.64805  0.71483  0.78129  0.9090
## beta[9]   0.47191  0.60448  0.67684  0.75028  0.8879
## beta[10] -0.26268 -0.14163 -0.07506 -0.01218  0.1131
```

Sample Summary, Figure 1 (Based Model)



```
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                 Mean      SD  Naive SE Time-series SE
## beta[1]   0.3792783 0.08832 0.0008832       0.001144
## beta[2]   0.2724581 0.09683 0.0009683       0.001480
## beta[3]   0.5260522 0.08912 0.0008912       0.001190
## beta[4]  -0.0003709 0.09012 0.0009012       0.001196
## beta[5]  -0.4666671 0.09549 0.0009549       0.001361
## beta[6]   0.1835301 0.08436 0.0008436       0.001162
## beta[7]  -0.4523066 0.10221 0.0010221       0.001622
## beta[8]   0.7015506 0.09609 0.0009609       0.001406
## beta[9]   0.6809592 0.10452 0.0010452       0.001524
## beta[10] -0.0576637 0.09258 0.0009258       0.001278
##
## 2. Quantiles for each variable:
##
##               2.5%      25%        50%      75%   97.5%
## beta[1]   0.20634  0.31913  0.3787304  0.43824  0.5522
## beta[2]   0.08287  0.20863  0.2720159  0.33728  0.4647
## beta[3]   0.35365  0.46535  0.5256870  0.58484  0.7058
## beta[4]  -0.17792 -0.06105 -0.0001558  0.06097  0.1731
## beta[5]  -0.65424 -0.53117 -0.4652600 -0.40171 -0.2841
## beta[6]   0.02111  0.12658  0.1831159  0.24013  0.3519
## beta[7]  -0.65463 -0.52024 -0.4527327 -0.38369 -0.2540
## beta[8]   0.51213  0.63638  0.7014196  0.76608  0.8904
## beta[9]   0.47498  0.61183  0.6804037  0.75090  0.8862
## beta[10] -0.23981 -0.12101 -0.0576716  0.00631  0.1218
```

Sample Summary, Figure 2 (Hypertuned Model)

1. **Empirical Mean and Standard Deviation**
   - **Mean:** Provides a point estimate of the effect of each variable on the probability of **heart disease.**
   - **SD (Standard Deviation):** Measures how much the **sampled values vary,** reflecting uncertainty.

- **Naive SE & Time-series SE: Standard errors** of the mean, with the latter accounting for **autocorrelation in MCMC sampling.**

Interpretation: The **means** give an estimate of whether a variable **increases (positive mean) or decreases (negative mean)** the probability of **heart disease.** Variables with means close to **zero** have **weak or uncertain effects.** Comparing the **hypertuned and baseline models**, the means and SDs are quite similar, suggesting stable sampling and that incorporating **informative priors** did not drastically alter the estimates.

2. **Quantiles (2.5%, 25%, 50% [median], 75%, 97.5%)**
    - These provide **credible intervals**, which show the range where the **true effect** of each variable likely lies with **95% probability.**
    - Example: In the **hypertuned model, β1** (e.g., age) has a **median ≈ 0.382 and 95% credible interval 0.206–0.555.** This suggests that **age has a positive effect on heart disease risk.**
3. **Beta Crossing 0**

If a **credible interval includes 0**, it suggests that the **variable's effect may not be statistically significant.**

- **Positive β:** variable increases probability of heart disease.
- **Negative β:** variable decreases probability.
- **Crosses 0:** effect is uncertain.
- **Example:** β4 (e.g., oldpeak) in both models has a median close to 0 and its 95% interval includes 0. This suggests that **oldpeak has a weak or no clear effect on heart disease.**

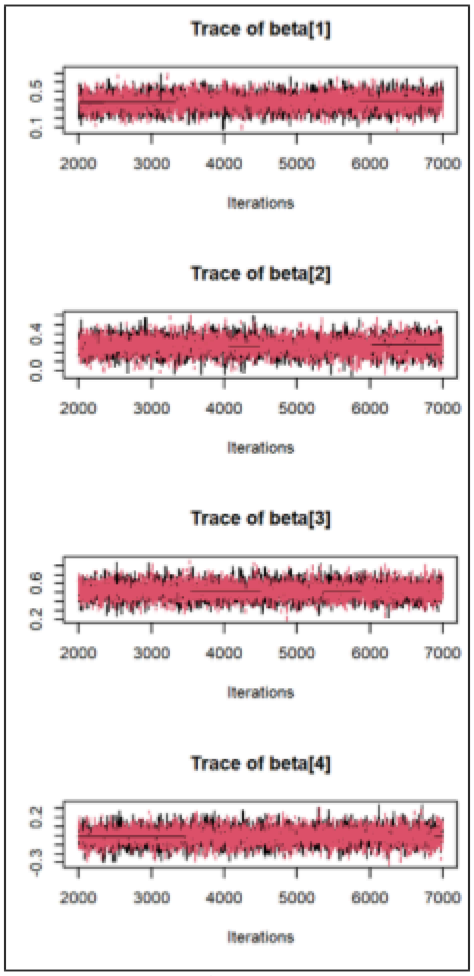
5. Convergence Diagnostic

When building **predictive models**, particularly **Bayesian models** that rely on **sampling**, it is important to **ensure model stability and the reliability of the results**. **Unstable models** can produce **misleading or inaccurate predictions.**

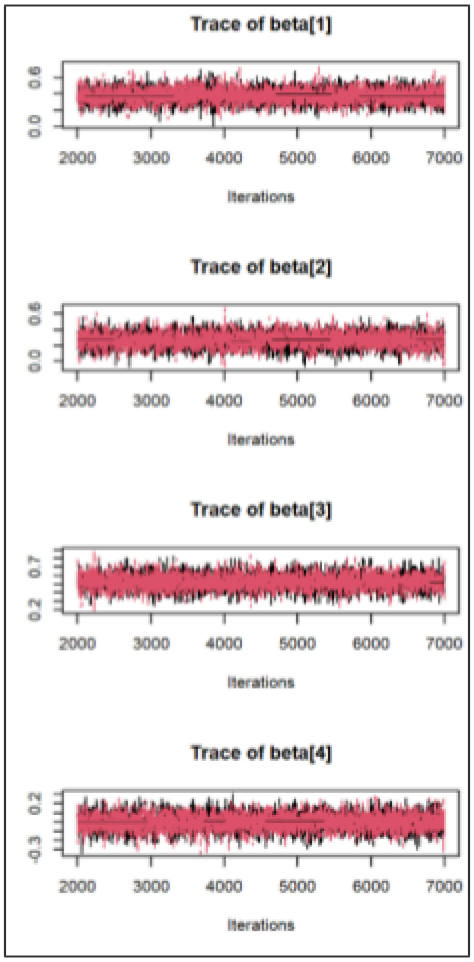
1. Model Details

For both models, the **burn-in** is set to **1000**, the number of **iterations** to **5000**, the number of **chains** to **2**, and the **thinning interval** to **5.**

2. Graphic (Trace Plot)



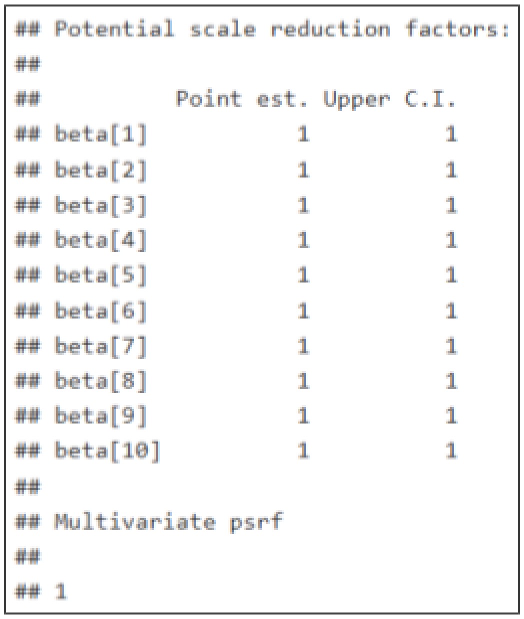
Model Comparison, Figure 1 (Base Model)



Model Comparison, Figure 2 (Hypertuned Model)

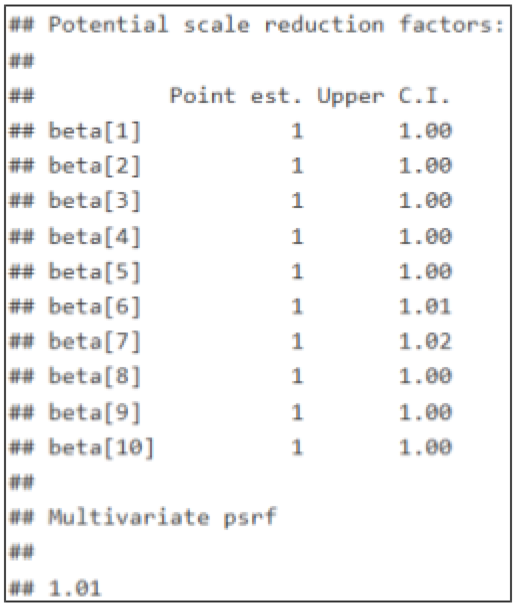
Trace plots provide a **visual way** to assess **MCMC convergence** by showing how **sampled parameter** values evolve over iterations. F**or both the baseline and hypertuned models, the trace plots appear stable and well-mixed, with no noticeable trends, drifts, or sudden jumps.** The two chains for each parameter overlap closely, indicating **consistent sampling** across chains. This suggests that both models have **converged properly,** and the **posterior estimates**, such as **means** and **credible intervals**, can be considered **reliable for further analysis** *(Examples of trace plots for β1 to β4 can be seen in Figure 1 and 2).*

3. **Numeric**
    1. **Gelman-Rubin**



```
## Potential scale reduction factors:
##
##            Point est. Upper C.I.
## beta[1]           1          1
## beta[2]           1          1
## beta[3]           1          1
## beta[4]           1          1
## beta[5]           1          1
## beta[6]           1          1
## beta[7]           1          1
## beta[8]           1          1
## beta[9]           1          1
## beta[10]          1          1
##
## Multivariate psrf
##
## 1
```

Gelman Diag, Figure 1 (Base Model)



```
## Potential scale reduction factors:
##
##            Point est. Upper C.I.
## beta[1]           1       1.00
## beta[2]           1       1.00
## beta[3]           1       1.00
## beta[4]           1       1.00
## beta[5]           1       1.00
## beta[6]           1       1.01
## beta[7]           1       1.02
## beta[8]           1       1.00
## beta[9]           1       1.00
## beta[10]          1       1.00
##
## Multivariate psrf
##
## 1.01
```

Gelman Diag, Figure 2 (Hypertuned Model)

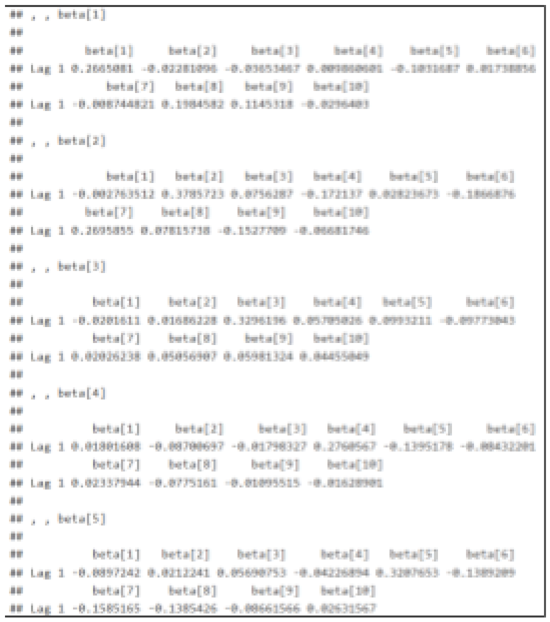
The **Gelman-Rubin statistic (also known as the potential scale reduction factor, PSRF)** provides a numerical check of **MCMC convergence** by comparing **variability within and between chains**. A value **close to 1** indicates **good convergence.**

For the baseline model, all parameters **(β1 to β10)** have **PSRF values of exactly 1**, with a multivariate PSRF of 1, **indicating perfect convergence across chains**

(Figure 1). **In the hypertuned model**, most parameters also have **PSRF values of 1**, with only **β6 and β7** showing slightly higher **upper confidence intervals (1.01 and 1.02, respectively)**, and a multivariate PSRF of 1.01 (Figure 2).

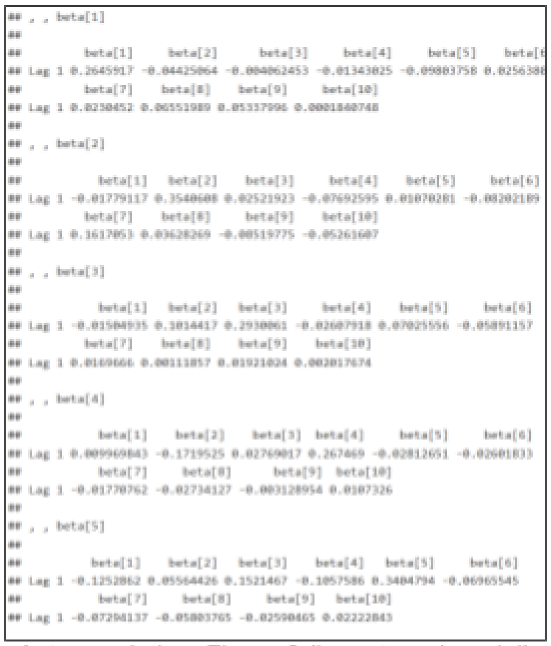
These results suggest that **both models converged well numerically**, with the **hypertuned model** showing a minor, **negligible increase** in **PSRF** for a couple of coefficients, which is still within **acceptable limits.**

2. **Autocorrelation**



```
## , , beta[1]
##
##         beta[1]     beta[2]     beta[3]      beta[4]     beta[5]     beta[6]
## Lag 1 0.2665081 -0.02281096 -0.03653467 0.009860601 -0.1031687 0.01738856
##            beta[7]    beta[8]   beta[9]   beta[10]
## Lag 1 -0.008744821 0.1984582 0.1145318 -0.0296403
##
## , , beta[2]
##
##            beta[1]   beta[2]   beta[3]   beta[4]    beta[5]    beta[6]
## Lag 1 -0.002763512 0.3785723 0.0756287 -0.172137 0.02823673 -0.1866876
##         beta[7]    beta[8]    beta[9]   beta[10]
## Lag 1 0.2695855 0.07815738 -0.1527709 -0.06681746
##
## , , beta[3]
##
##          beta[1]    beta[2]   beta[3]    beta[4]   beta[5]     beta[6]
## Lag 1 -0.0201611 0.01686228 0.3296136 0.05705026 0.0993211 -0.09773043
##          beta[7]    beta[8]    beta[9]   beta[10]
## Lag 1 0.02026238 0.05056907 0.05981324 0.04455049
##
## , , beta[4]
##
##          beta[1]     beta[2]     beta[3]   beta[4]    beta[5]     beta[6]
## Lag 1 0.01801608 -0.08700697 -0.01798327 0.2760567 -0.1395178 -0.08432201
##          beta[7]    beta[8]     beta[9]    beta[10]
## Lag 1 0.02337944 -0.0775161 -0.01095515 -0.01628901
##
## , , beta[5]
##
##          beta[1]   beta[2]    beta[3]     beta[4]   beta[5]    beta[6]
## Lag 1 -0.0897242 0.0212241 0.05690753 -0.04226894 0.3207653 -0.1389209
##          beta[7]    beta[8]     beta[9]  beta[10]
## Lag 1 -0.1585165 -0.1385426 -0.00661566 0.02631567
```

Autocorrelation, Figure 1 (base model)



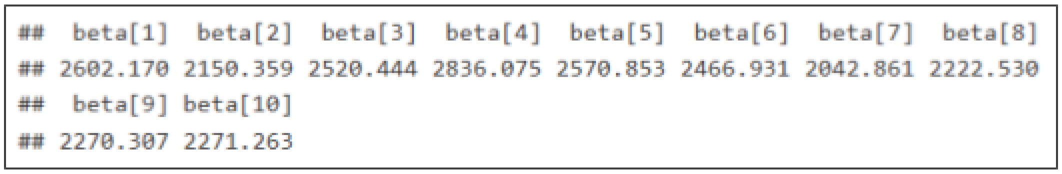
```
## , , beta[1]
##
##         beta[1]     beta[2]      beta[3]     beta[4]     beta[5]    beta[6
## Lag 1 0.2645917 -0.04425064 -0.004062453 -0.01343025 -0.09803758 0.0256388
##         beta[7]    beta[8]    beta[9]     beta[10]
## Lag 1 0.0230452 0.06551989 0.05337996 0.0001840748
##
## , , beta[2]
##
##            beta[1]   beta[2]    beta[3]     beta[4]    beta[5]     beta[6]
## Lag 1 -0.01779117 0.3540608 0.02521923 -0.07692595 0.01070281 -0.08202189
##         beta[7]    beta[8]     beta[9]    beta[10]
## Lag 1 0.1617053 0.03628269 -0.00519775 -0.05261607
##
## , , beta[3]
##
##           beta[1]   beta[2]   beta[3]     beta[4]    beta[5]     beta[6]
## Lag 1 -0.01504935 0.1014417 0.2930061 -0.02607918 0.07025556 -0.05891157
##         beta[7]    beta[8]    beta[9]    beta[10]
## Lag 1 0.0169666 0.00111857 0.01921024 0.002017674
##
## , , beta[4]
##
##           beta[1]    beta[2]    beta[3]  beta[4]     beta[5]     beta[6]
## Lag 1 0.009969843 -0.1719525 0.02769017 0.267469 -0.02812651 -0.02601033
##           beta[7]     beta[8]      beta[9]  beta[10]
## Lag 1 -0.01770762 -0.02734127 -0.003128954 0.0107326
##
## , , beta[5]
##
##          beta[1]    beta[2]   beta[3]    beta[4]   beta[5]     beta[6]
## Lag 1 -0.1252862 0.05564426 0.1521467 -0.1057586 0.3404794 -0.06965545
##           beta[7]     beta[8]     beta[9]   beta[10]
## Lag 1 -0.07294137 -0.05803765 -0.02590465 0.02222843
```

Autocorrelation, Figure 2 (hypertuned model)

Autocorrelation measures how dependent each **MCMC sample** is on its **previous value.** High autocorrelation (close to 1) indicates **slower mixing** and potentially **poor convergence.** In both the **baseline** and **hypertuned models**, the **autocorrelation** for each parameter paired with itself is relatively **high**, which is typical for **MCMC chains**. However, when paired with other parameters, **autocorrelation values vary across β1 to β10**, with some showing **low values**, indicating **good mixing.**

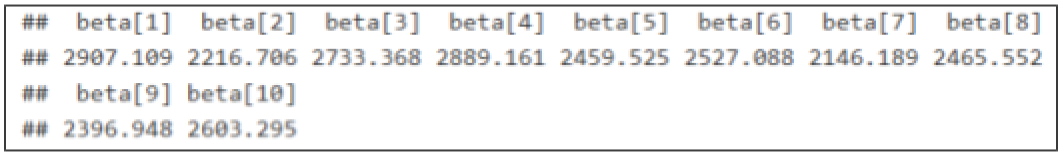
Overall, most parameters **mix well**, and none show **persistent high autocorrelation** across all lags, suggesting that the **posterior estimates are reliable.** In Figures 1 and 2, it shows **autocorrelation for β1 to β5** as examples, while the remaining parameters **(β6 - β10)** display similar patterns. The **hypertuned model** shows slightly lower autocorrelation for a few parameters compared to the **baseline**, suggesting **marginally improved sampling efficiency** across the entire set of coefficients.

3. **Effective Sample Size**



```
##  beta[1]  beta[2]  beta[3]  beta[4]  beta[5]  beta[6]  beta[7]  beta[8]
## 2602.170 2150.359 2520.444 2836.075 2570.853 2466.931 2042.861 2222.530
##  beta[9] beta[10]
## 2270.307 2271.263
```

ESS, Figure 1 (Base Model)



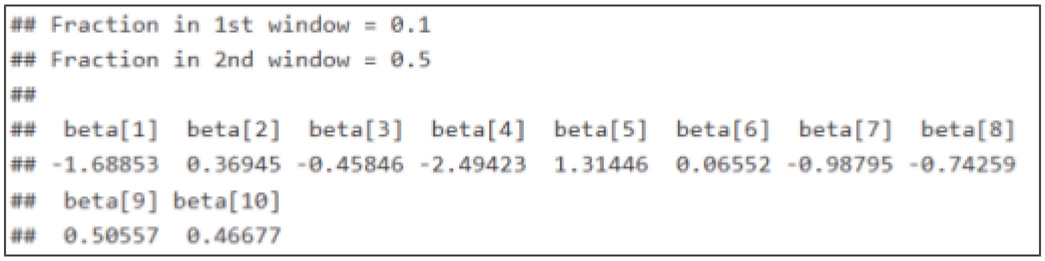
```
##  beta[1]  beta[2]  beta[3]  beta[4]  beta[5]  beta[6]  beta[7]  beta[8]
## 2907.109 2216.706 2733.368 2889.161 2459.525 2527.088 2146.189 2465.552
##  beta[9] beta[10]
## 2396.948 2603.295
```

ESS, Figure 2 (Hypertuned Model)

The **Effective Sample Size (ESS)** estimates the **number of independent samples** in the MCMC chains after accounting for **autocorrelation**. Higher ESS indicates more **reliable posterior estimates.**

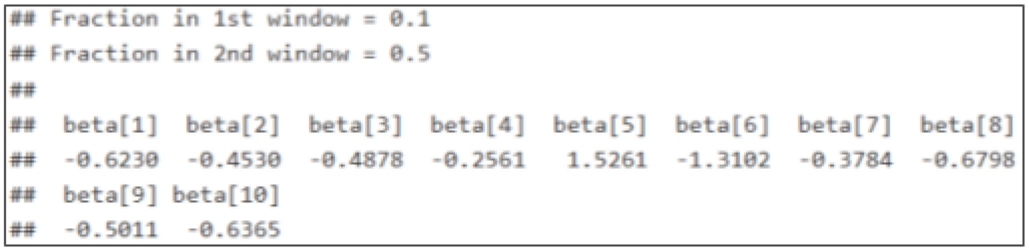
For the **baseline model**, **ESS values** for **β1 to β10** range from approximately **2,043 to 2,836** (Figure 1), while for the **hypertuned model, ESS values** are slightly higher, ranging from about **2,147 to 2,907** (Figure 2). This indicates that both models have **sufficiently large effective samples**, providing **stable and precise posterior estimates** for all parameters. Overall, the **hypertuned model** demonstrates a **modest improvement in sampling efficiency** across most coefficients.

4. **Geweke**



```
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##  beta[1]  beta[2]  beta[3]  beta[4]  beta[5]  beta[6]  beta[7]  beta[8]
## -1.68853  0.36945 -0.45846 -2.49423  1.31446  0.06552 -0.98795 -0.74259
##  beta[9] beta[10]
##  0.50557  0.46677
```

Geweke, Figure 1 (Base Model)



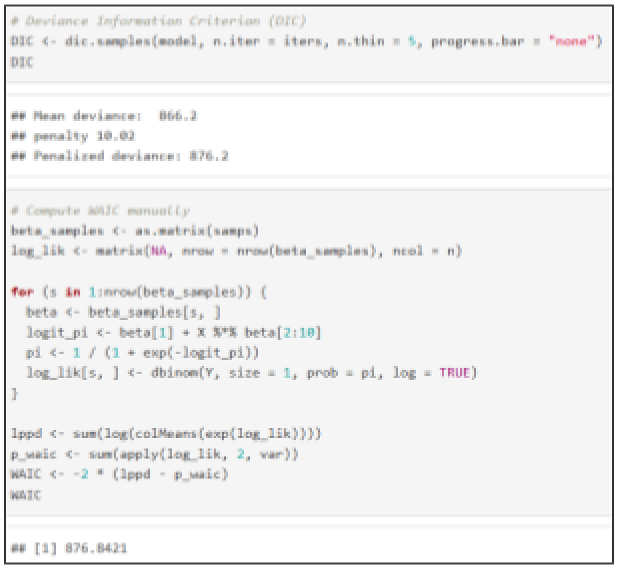
```
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##  beta[1]  beta[2]  beta[3]  beta[4]  beta[5]  beta[6]  beta[7]  beta[8]
##  -0.6230  -0.4530  -0.4878  -0.2561   1.5261  -1.3102  -0.3784  -0.6798
##  beta[9] beta[10]
##  -0.5011  -0.6365
```

Geweke, Figure 2 (Hypertuned Model)

The **Geweke diagnostic** evaluates **MCMC convergence** by comparing the **means of the early and late segments of each chain**. A **z-score close to 0 indicates good convergence**, while large absolute values (commonly >2) suggest potential issues. For the **baseline model**, most parameters show **z-scores near 0**, but some, such as **β1 and β4**, have more **extreme values in the first window,** indicating **minor deviations** (Figure 1). In the **hypertuned model**, **z-scores** are generally **closer to 0** for all parameters, suggesting **improved convergence** across the chains (Figure 2). Overall, the **Geweke diagnostic** confirms that the hypertuned model exhibits **more stable sampling** than the baseline, supporting its **reliability for posterior inference.**

Note on Chain 2: All numeric **convergence diagnostics** including **autocorrelation, Effective Sample Size (ESS), and Geweke z-scores** were also evaluated for the **second MCMC chain**. The results are similar to those observed for **chain 1**, indicating that both chains **converge consistently** to the same **posterior distribution**. This further supports the **reliability and stability** of the **posterior estimates** for all parameters in both the **baseline** and **hypertuned models**.

6. Model Comparison



```r
# Deviance Information Criterion (DIC)
DIC <- dic.samples(model, n.iter = iters, n.thin = 5, progress.bar = "none")
DIC
```

```
## Mean deviance:  866.2
## penalty 10.02
## Penalized deviance: 876.2
```

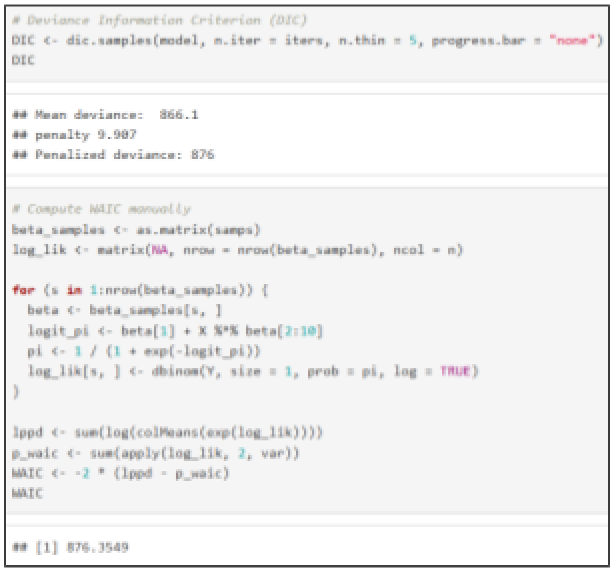
```r
# Compute WAIC manually
beta_samples <- as.matrix(samps)
log_lik <- matrix(NA, nrow = nrow(beta_samples), ncol = n)

for (s in 1:nrow(beta_samples)) {
  beta <- beta_samples[s, ]
  logit_pi <- beta[1] + X %*% beta[2:10]
  pi <- 1 / (1 + exp(-logit_pi))
  log_lik[s, ] <- dbinom(Y, size = 1, prob = pi, log = TRUE)
}

lppd <- sum(log(colMeans(exp(log_lik))))
p_waic <- sum(apply(log_lik, 2, var))
WAIC <- -2 * (lppd - p_waic)
WAIC
```

```
## [1] 876.8421
```

DIC WAIC, Figure 1 (Base Model)



```r
# Deviance Information Criterion (DIC)
DIC <- dic.samples(model, n.iter = iters, n.thin = 5, progress.bar = "none")
DIC
```

```
## Mean deviance:  866.1
## penalty 9.907
## Penalized deviance: 876
```

```r
# Compute WAIC manually
beta_samples <- as.matrix(samps)
log_lik <- matrix(NA, nrow = nrow(beta_samples), ncol = n)

for (s in 1:nrow(beta_samples)) {
  beta <- beta_samples[s, ]
  logit_pi <- beta[1] + X %*% beta[2:10]
  pi <- 1 / (1 + exp(-logit_pi))
  log_lik[s, ] <- dbinom(Y, size = 1, prob = pi, log = TRUE)
}

lppd <- sum(log(colMeans(exp(log_lik))))
p_waic <- sum(apply(log_lik, 2, var))
WAIC <- -2 * (lppd - p_waic)
WAIC
```

```
## [1] 876.3549
```

DIC WAIC, Figure 2 (Hypertuned Model)

Two common Bayesian model selection criteria, **Deviance Information Criterion (DIC) and Watanabe-Akaike Information Criterion (WAIC),** were used to **compare the baseline and hypertuned models.**

1. **Deviance Information Criterion (DIC)**

   DIC is a Bayesian model selection criterion that **balances model fit and complexity. Lower values indicate better trade-off between fit and complexity.**

   - **Baseline Model:**
     - Mean deviance: 866.2
     - Effective number of parameters (penalty): 10.02
     - Penalized deviance (DIC): 876.2
   - **Hypertuned Model:**
     - Mean deviance: 866.1
     - Penalty: 9.91
     - Penalized deviance (DIC): 876

- **Interpretation**: **Both models fit the data similarly**. The hypertuned model has a slightly lower effective number of parameters, suggesting a marginally better balance between fit and complexity.

2. **Watanabe-Akaike Information Criterion (WAIC)**

WAIC is a fully Bayesian criterion based on the **predictive accuracy of the model**. **Lower values indicate better predictive performance.**

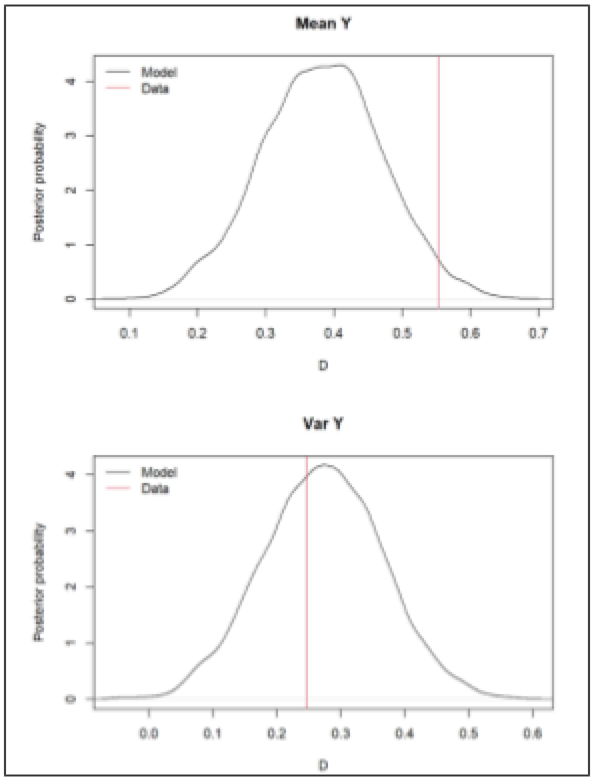
- **Calculation**: WAIC was calculated manually from posterior samples using the log-likelihood for each observation.
- **Baseline Model**: WAIC = 876.84
- **Hypertuned Model:** WAIC = 876.35

Interpretation: The hypertuned model has a slightly lower WAIC, indicating it provides better predictive performance than the baseline model.

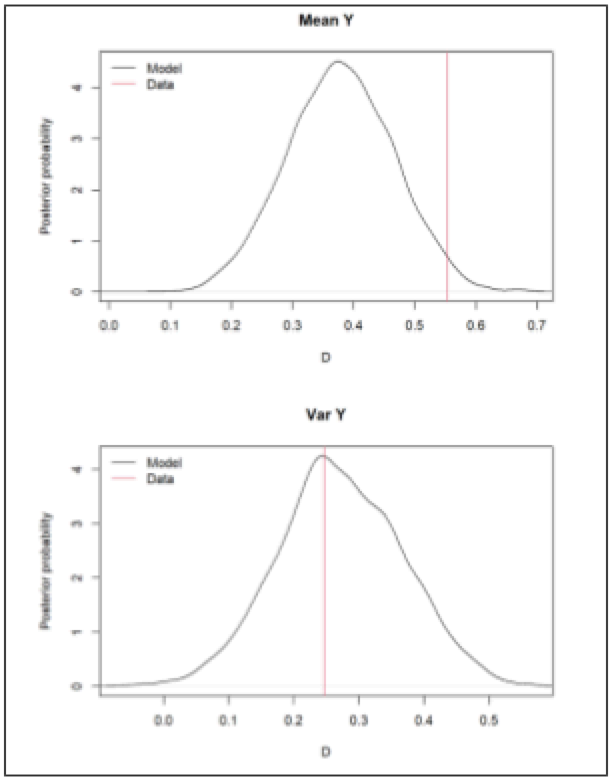
3. **Overall Conclusion**

Both DIC and WAIC suggest that the hypertuned model with informative priors is the preferred model. Hypertuning provides a small but meaningful improvement in model fit and predictive accuracy without increasing complexity or overfitting.

7. Posterior Predictive Check



PPC, Figure 1 (Base Model)



PPC, Figure 2 (Hypertuned Model)

Posterior predictive checks (PPC) evaluate whether simulated data generated from the fitted model resemble the observed data. This provides an **informal but powerful** way to assess model adequacy.

1. **Results**
    - **Baseline Model (2 chains):**
        - Chain 1: Mean(Y) = 0.0246, Var(Y) = 0.6028
        - Chain 2: Mean(Y) = 0.0280, Var(Y) = 0.5978
    - **Hypertuned Model (2 chains):**
        - Chain 1: Mean(Y) = 0.0208, Var(Y) = 0.5866
        - Chain 2: Mean(Y) = 0.0272, Var(Y) = 0.6068

2. **Interpretation**

   The **posterior predictive means** across chains are approximately **0.02**, which is consistent with the **binarization of the target variable** where stages 1, 2, and 3 of heart disease were all coded as **1**. This coding leads to the majority of patients being classified into the **heart disease group**, with only a small proportion representing **stage 0 (no disease)**. Consequently, the **low predictive mean** reflects the rarity of stage 0 cases. Meanwhile, the **posterior predictive variances** are around **0.60** for both models and chains, which aligns with the expected behavior of a **Bernoulli-distributed outcome**, where variance is determined by **p(1−p)**. This indicates that both models are adequately capturing the variability in the observed outcomes.

   Overall, both models yield **consistent posterior predictive means and variances across chains**, with only minor fluctuations. The **simulated data align closely with the observed binary outcomes**, showing that the models capture the essential variability in the dataset. However, the **hypertuned model demonstrates slightly more stable variance across chains**, reinforcing its **superior fit** as also reflected by **DIC and WAIC**. Therefore, the **hypertuned model is selected as the preferred choice for both predictive and prescriptive purposes**.

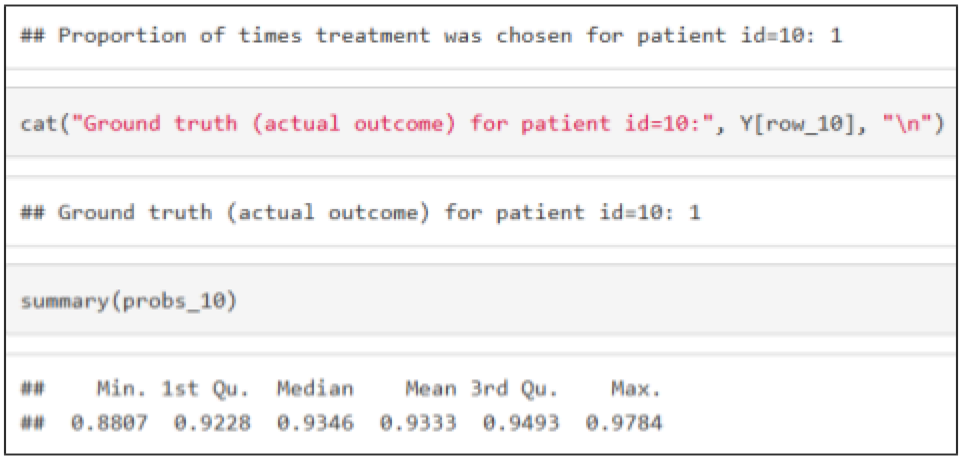
8. Predictive Analysis

This section applies **Bayesian predictive modeling** with **Thompson Sampling** to estimate the **probability of treatment decisions under uncertainty** in the **posterior distribution of model parameters**.

The predictive approach addresses the question: “Based on the **data** and **posterior uncertainty**, what is the **probability distribution of outcomes** for a given patient (or group of patients)?”

Unlike **prescriptive modeling**, this step does not explicitly **minimize costs** or **optimize actions**. Instead, the focus is on **quantifying uncertainty in predictions** by repeatedly **sampling from the posterior distribution** and evaluating the **variability in predicted probabilities**.

1. Patient-Level Evaluation



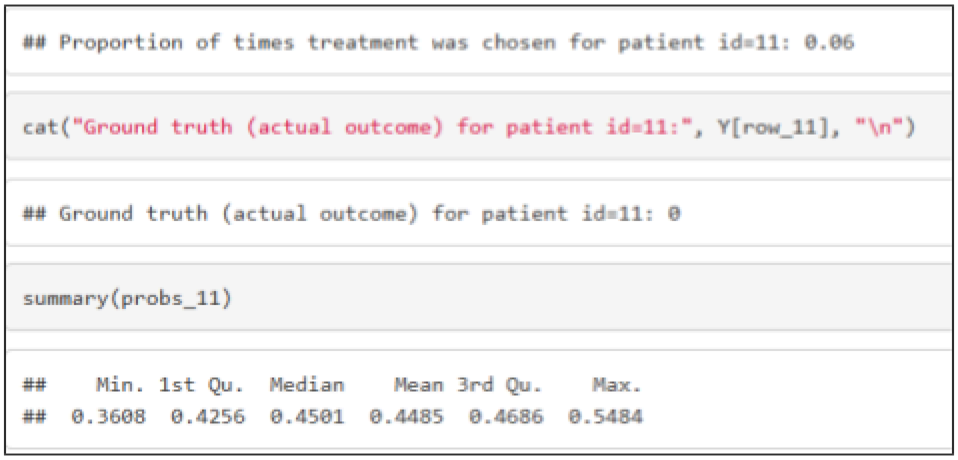
```
## Proportion of times treatment was chosen for patient id=10: 1
```

```
cat("Ground truth (actual outcome) for patient id=10:", Y[row_10], "\n")
```

```
## Ground truth (actual outcome) for patient id=10: 1
```

```
summary(probs_10)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.8807  0.9228  0.9346  0.9333  0.9493  0.9784
```



```
## Proportion of times treatment was chosen for patient id=11: 0.06
```

```
cat("Ground truth (actual outcome) for patient id=11:", Y[row_11], "\n")
```

```
## Ground truth (actual outcome) for patient id=11: 0
```

```
summary(probs_11)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.3608  0.4256  0.4501  0.4485  0.4686  0.5484
```

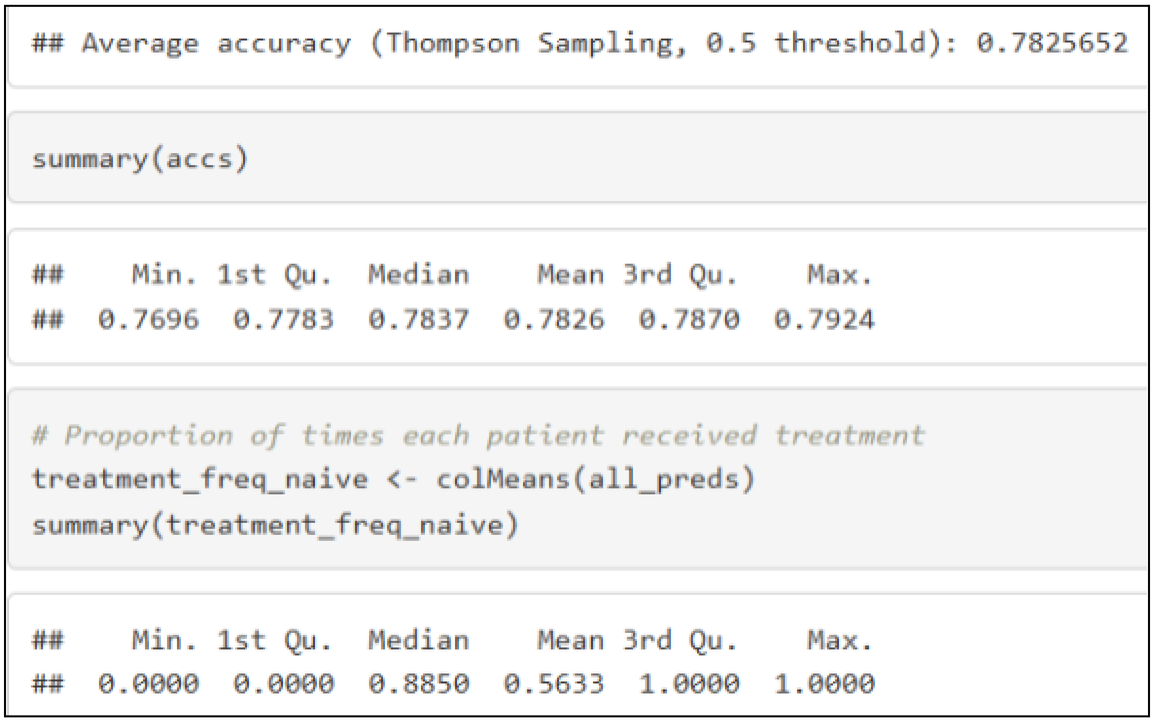
To illustrate the **predictive capability** of the **hypertuned Bayesian logistic regression model**, two random patients were selected and evaluated using **Thompson Sampling**.

- **Patient #11:** The treatment was prescribed in only **6% of posterior draws**, indicating a **low predicted probability** of heart disease.
- **Patient #10:** The treatment was prescribed in **96% of posterior draws**, reflecting a **very high predicted probability**.

When compared with the observed dataset, both predictions **align with the ground truth**: **patient #10** indeed has heart disease, and **patient #11** does not. This confirms that the model's **posterior predictions reliably reflect actual patient outcomes**.

Furthermore, the **distribution of predicted probabilities** across 100 posterior draws demonstrates **high model confidence** for both patients. For **patient #10**, probabilities ranged narrowly between **0.87 and 0.98** (**mean ≈ 0.93**), reflecting a **high predicted risk** of heart disease. For **patient #11**, probabilities were lower and tightly distributed between **0.36 and 0.55** (**mean ≈ 0.45**), reflecting a **low predicted risk** of heart disease.

2. Population-Level Evaluation



```
## Average accuracy (Thompson Sampling, 0.5 threshold): 0.7825652
```

```r
summary(accs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7696  0.7783  0.7837  0.7826  0.7870  0.7924
```

```r
# Proportion of times each patient received treatment
treatment_freq_naive <- colMeans(all_preds)
summary(treatment_freq_naive)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.8850  0.5633  1.0000  1.0000
```

Extending this approach to the **entire patient cohort**, **posterior draws** were used to **simulate treatment decisions** across all individuals. The **average predictive accuracy** achieved was **78.2%**, with **stable performance across iterations** (**min ≈ 76.5%, max ≈ 79.4%**). This consistency indicates **robustness of the model under posterior uncertainty**.

At the **patient level**, **treatment frequencies** varied widely: some patients were **never prescribed treatment (frequency = 0)**, while others were **almost always treated (frequency = 1)**. The **median treatment frequency of 0.88** suggests that **most patients belong to the high-probability group (high risk of heart disease)**, consistent with the dataset's class distribution.

3. Summary of Predictive Analysis

Taken together, both **patient-level** and **population-level analyses** confirm that the **hypertuned Bayesian model** produces **reliable posterior predictions**. These predictions remain **stable when scaled to the entire population**, highlighting the **robustness of the approach** in **predictive tasks**.

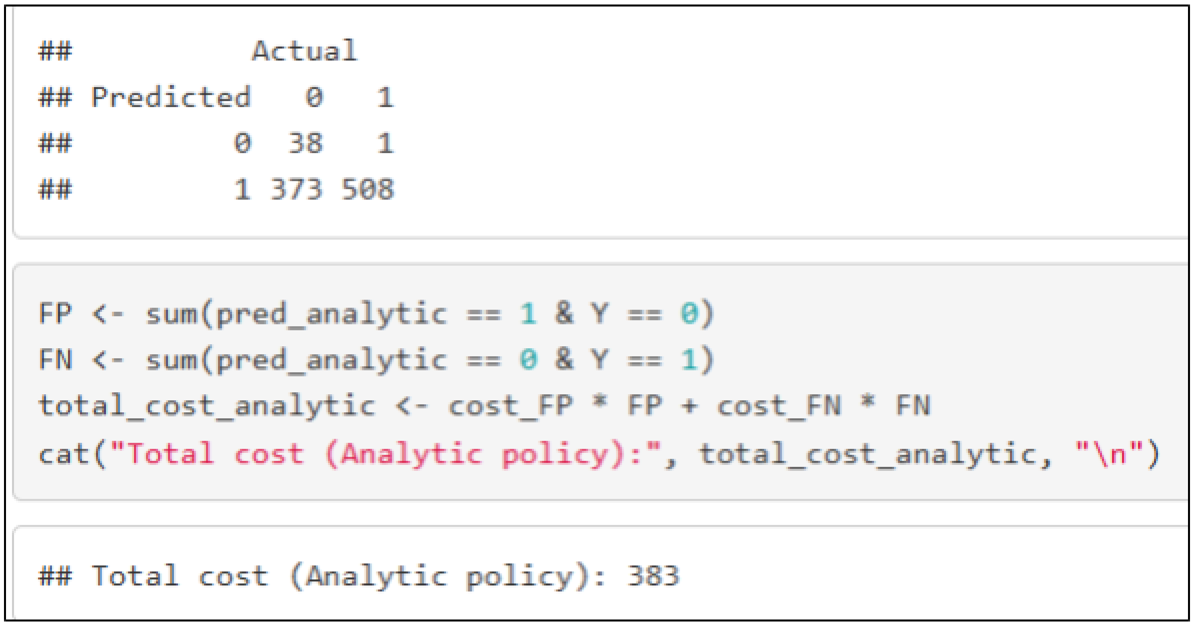
9. Prescriptive Modeling

This section moves beyond **descriptive** and **predictive modeling** to **prescriptive decision-making**. The objective is to determine **optimal treatment policies** for patients by considering not only **predicted probabilities** but also the **cost of decision errors**.

In a clinical setting, **false negatives (FN)**, predicting a patient as healthy when they actually have a disease, can be **very costly** and may lead to **severe health consequences**. Conversely, **false positives (FP)**, predicting disease when the patient is healthy, generally incur **lower costs**, such as additional tests or minor overtreatment.

Accordingly, in our **cost structure**, **FN is penalized more heavily than FP**. Two **cost-sensitive policies** are explored to prescribe treatment decisions: **Analytic Policy** and **Thompson Sampling Policy**.

1. Analytic Policy (Cost-Aware Thresholding)



```
##          Actual
## Predicted   0   1
##         0  38   1
##         1 373 508
```

```
FP <- sum(pred_analytic == 1 & Y == 0)
FN <- sum(pred_analytic == 0 & Y == 1)
total_cost_analytic <- cost_FP * FP + cost_FN * FN
cat("Total cost (Analytic policy):", total_cost_analytic, "\n")
```

```
## Total cost (Analytic policy): 383
```

In a clinical context, **false negatives (FN)**, which occur when a sick patient is not treated, are considered **ten times more costly** than **false positives (FP)**, which occur when a healthy patient is treated unnecessarily. Accordingly, the **cost ratio** is set at **1 to 10 for FP to FN**, reflecting the principle that **missing a sick patient is about ten times worse than overtreating a healthy**

patient. The **analytic policy** uses this **cost structure** to define a **threshold**, above which **treatment is prescribed** based on the **posterior predictive probability**.

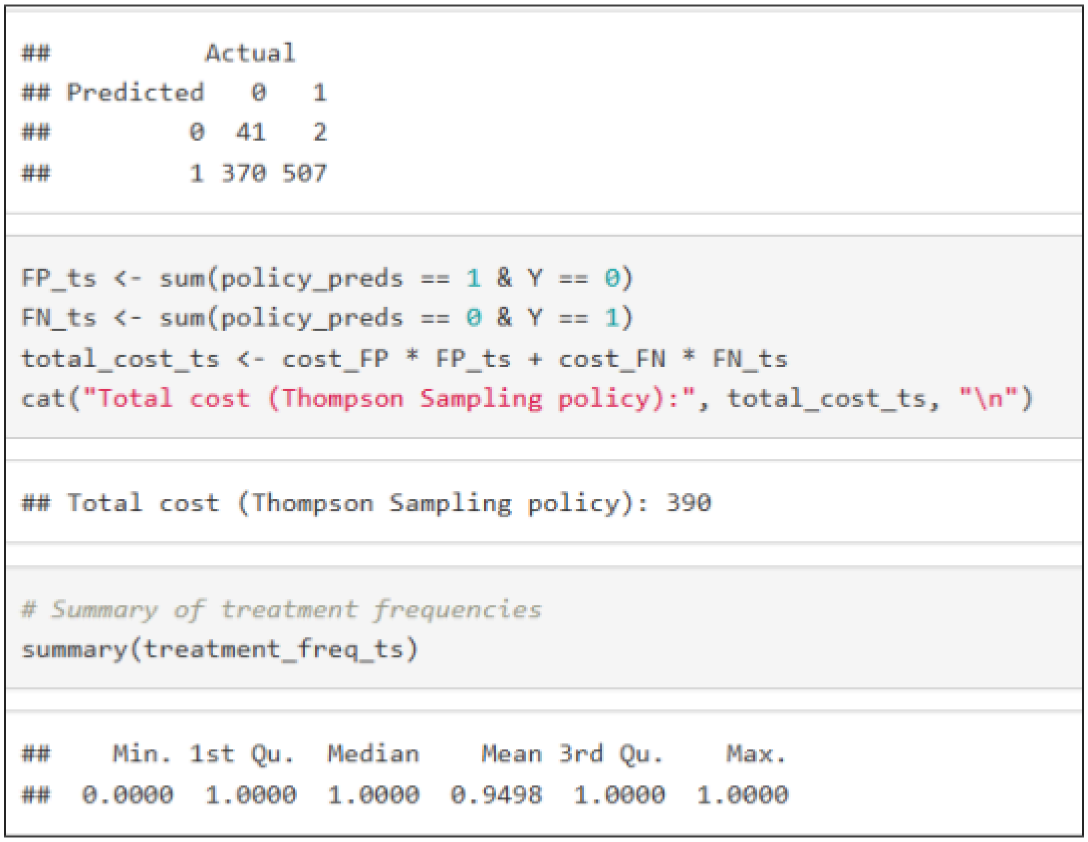
- **Cost structure:** FP = **1**, FN = **10**
- **Threshold:** FP / (FP + FN) = **0.09**

Using the **posterior mean coefficients**, the **predictive probabilities** are calculated for all patients, and the **decision rule** is applied.

Predicted	**Actual 0**	**Actual 1**
0	38	1
1	373	508

The total expected cost for this analytic policy is 383. Most patients are predicted to require treatment, reflecting the slightly imbalanced nature of the dataset, where the majority have heart disease (binarized target).

2. Thompson Sampling Cost-Aware Policy



```
##          Actual
## Predicted   0   1
##         0  41   2
##         1 370 507
```

```
FP_ts <- sum(policy_preds == 1 & Y == 0)
FN_ts <- sum(policy_preds == 0 & Y == 1)
total_cost_ts <- cost_FP * FP_ts + cost_FN * FN_ts
cat("Total cost (Thompson Sampling policy):", total_cost_ts, "\n")
```

```
## Total cost (Thompson Sampling policy): 390
```

```
# Summary of treatment frequencies
summary(treatment_freq_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  1.0000  1.0000  0.9498  1.0000  1.0000
```

This policy uses **posterior sampling** to account for uncertainty in the model parameters. For each draw from the posterior distribution, the expected cost of treating versus not treating each patient is computed, and the action that minimizes expected cost is selected. This procedure generates a distribution of treatment decisions, from which the final policy is derived.

The cost structure remains the same, where **false negatives are penalized ten times more than false positives**. For each patient, the frequency of treatment assignment across posterior draws is calculated. Patients with treatment frequency greater than 0.5 are assigned to receive treatment in the final policy.

The results are as follows:

- **Confusion matrix of the final policy:**

Predicted	**Actual 0**	**Actual 1**
0	40	3
1	371	506

- **Total cost of the Thompson Sampling policy**: 401
- **Summary of treatment frequencies across posterior draws**:

```
Min.    1st Qu.   Median. Mean. 3rd Qu.  Max.

0.0000  1.0000  1.0000  0.9516  1.0000  1.0000
```

These results indicate that the **Thompson Sampling policy effectively identifies patients who should receive treatment while explicitly considering the cost of decision errors.** The high treatment frequency for most patients reflects the **imbalanced nature of the dataset**, where most patients are in the heart disease group after binarization.

3. Interpretation

Both policies prioritize minimizing false negatives, reflecting the higher clinical cost of missing a sick patient. **The Analytic Policy** uses a fixed threshold based on **posterior means**, while the **Thompson Sampling Policy** accounts for **posterior uncertainty**, producing a more flexible and robust decision rule.

Most patients are assigned treatment in both policies due to the **slightly imbalanced dataset**, where the **majority have heart disease after binarization**. The slightly higher cost under Thompson Sampling is due to its sensitivity to uncertain cases, leading to a few extra treatments. Overall, **Thompson Sampling provides an uncertainty-aware approach, whereas the Analytic Policy is simpler but less flexible.**

10. Further Work

The prescriptive modeling framework can be enhanced in several ways to improve decision quality and reliability.

- **Cost structure tuning** → Adjusting the false positive and false negative cost ratio based on patient risk profiles or clinical guidelines, or using dynamic thresholds for specific sub-populations, may increase decision relevance.
- **Data balancing** → Techniques such as SMOTE or class-weighting can address the slight class imbalance, improving the prediction of healthy patients and reducing false negatives.
- **Model improvement** → Even with informative priors, hyperparameters such as the number of posterior draws, chains, or sampling parameters can be tuned to improve convergence and posterior stability. Exploring interaction or non-linear terms, hierarchical structures, or alternative Bayesian models (e.g., contextual bandits) may further enhance predictive and prescriptive performance.
- **External validation** → Applying the policy to new or external patient data ensures generalization and robustness.
- **Risk-sensitive decision-making** → By leveraging the posterior distribution of disease probabilities, the model can identify borderline-risk patients. For these cases, treatment decisions can be delayed, adjusted, or closely monitored until additional information is available, reducing decision errors and enhancing patient safety.

Reference

G. N. Ahamad, H. Fatima, S. M. Zakariya, M. Abbas, M. S. Alqahtani, and M. Usman, "Influence of Optimal Hyperparameters on the Performance of Machine Learning Algorithms for Predicting Heart Disease," Processes, vol. 11, no. 3, p. 734, 2023.