IntroToBayesianInference

Mahtab Nahayati

2024-12-24

Contents

Bayesian Estimation of Covid19 Prevalence 3

Posterior Distribution for Covid19 Prevalence 4

Posterior Analysis for Covid19 Prevalence 5

Discussion 7

Define Conjugate Priors for Bayesian Linear Modelling 7

Build the Normal Model for Regression Inference 8

Point Estimators and 95% HPD Interval for Regression Parameters 9

Bayesian vs Frequentist Regression Analysis 10

Bayesian Estimation of Covid19 Prevalence

We recalculate the estimation of the prevalence of Covid19 in spring 2020. Samples from 1279 persons were analysed with PCR testing procedures. Out of all those not a single randomly selected person was tested positively. This obviously breaks standard testing mechanisms for extimating the proportion of infected person in Austria.

However, additional information is available from similar tests in Germany which had a comparable behaviour of the spread of the disease at that time. In the same time span 4 positive cases out of 4068 had been found.

We will build a Beta prior distribution for this Bionomial scenario, which encodes the information of the German study.

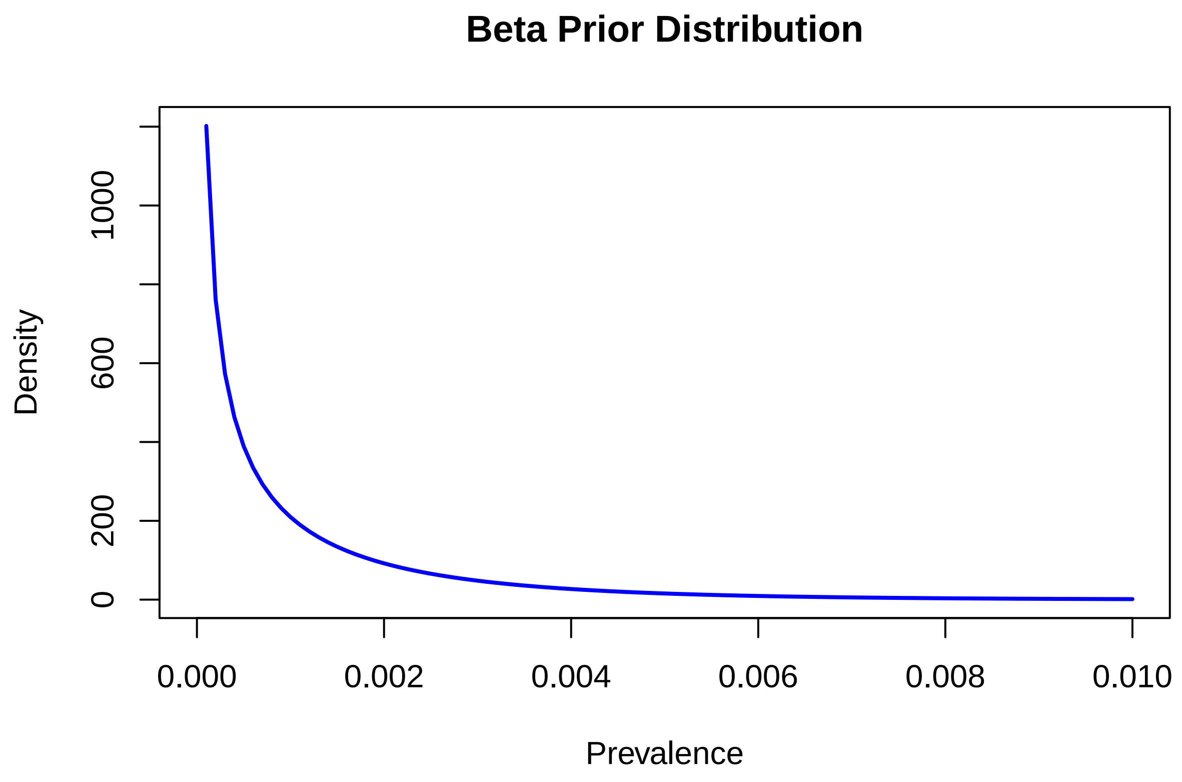
```
# Information from the German study
positive_cases_germany <- 4
total_tests_germany <- 4068

# Reweighting factor
reweight_factor <- 1/10

# Calculating prior parameters
alpha_prior <- positive_cases_germany * reweight_factor
beta_prior <- (total_tests_germany - positive_cases_germany) * reweight_factor

# Display the prior parameters
cat("Alpha (a):", alpha_prior, "\n")
```

```
## Alpha (a): 0.4
```

```
cat("Beta (b):", beta_prior, "\n")
```

```
## Beta (b): 406.4
```

```
# Plotting the Beta prior distribution
curve(dbeta(x, alpha_prior, beta_prior), from = 0, to = 0.01,
      main = "Beta Prior Distribution",
      xlab = "Prevalence",
      ylab = "Density",
      col = "blue", lwd = 2)
```



Posterior Distribution for Covid19 Prevalence

We want to build the corresponding Binomial model for the number of people suffering from the disease based on the 1279 test.We will obtain the theoretical posterior distribution for this scenario.

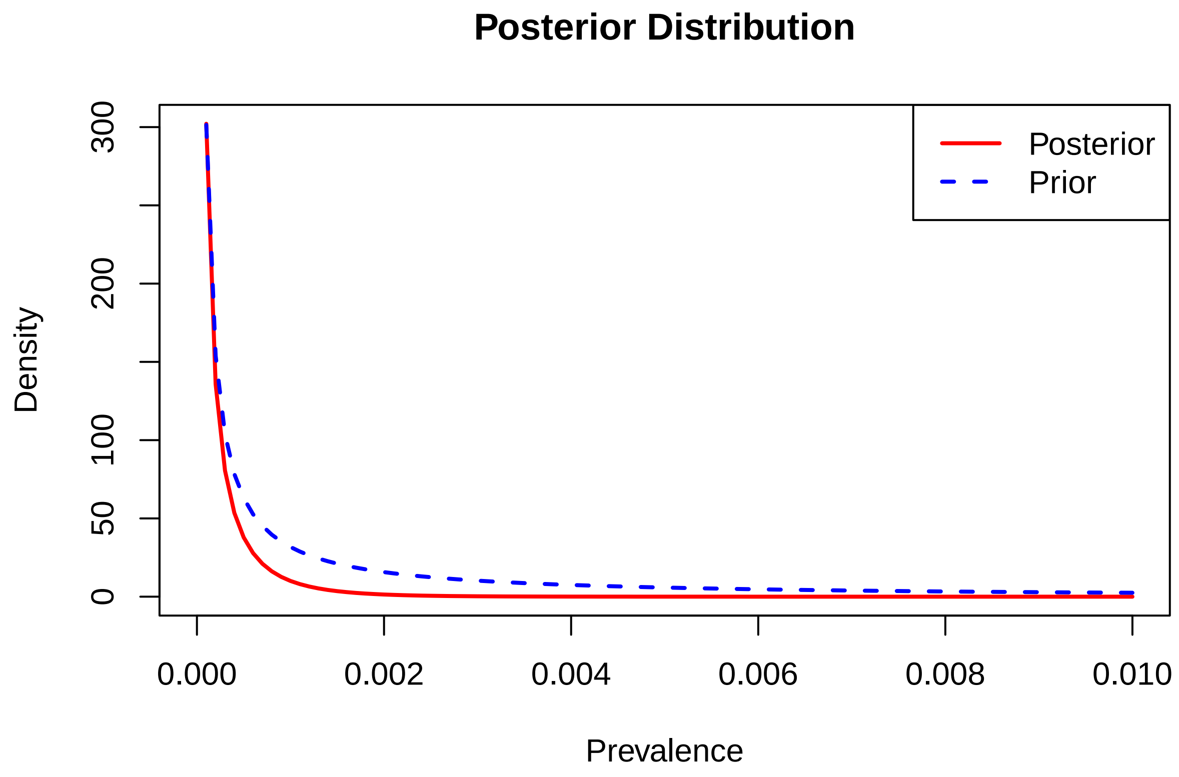
```
# Information from the Austrian test
positive_cases_austria <- 0
total_tests_austria <- 1279

# Prior parameters from the German study
reweight_factor <- 110 # Reweight factor as per Task 1.1
alpha_prior <- 4 * (1 / reweight_factor)  # Reweighted alpha
beta_prior <- (4068 - 4) * (1 / reweight_factor)  # Reweighted beta

# Posterior parameters
alpha_posterior <- alpha_prior + positive_cases_austria
beta_posterior <- beta_prior + (total_tests_austria - positive_cases_austria)

# Display the posterior parameters
cat("Alpha (Posterior a):", alpha_posterior, "\n")
```

```
## Alpha (Posterior a): 0.03636364
```

```
cat("Beta (Posterior b):", beta_posterior, "\n")
```

```
## Beta (Posterior b): 1315.945
```

```r
# Plotting the posterior distribution
curve(dbeta(x, alpha_posterior, beta_posterior), from = 0, to = 0.01,
      main = "Posterior Distribution",
      xlab = "Prevalence",
      ylab = "Density",
      col = "red", lwd = 2)

# Add prior distribution for comparison
curve(dbeta(x, alpha_prior, beta_prior), from = 0, to = 0.01,
      col = "blue", lwd = 2, lty = 2, add = TRUE)

# Add legend
legend("topright", legend = c("Posterior", "Prior"),
       col = c("red", "blue"), lwd = 2, lty = c(1, 2))
```



Posterior Analysis for Covid19 Prevalence

We will plot the posterior density and obtain the point estimators and 95% Highest posterior density interval of the prevalence of Covid19 (=proportion of inhabitants suffering from the disease).

```r
# Load necessary libraries
library(coda) # For HPD interval
```

```
## Warning: Paket 'coda' wurde unter R Version 4.3.3 erstellt
```

```r
# Define prior parameters
alpha_prior <- 1.43
beta_prior <- 1116.57

# Define posterior parameters
alpha_posterior <- alpha_prior + 0 # No positives in the Austrian sample
beta_posterior <- beta_prior + 1279 # Total Austrian sample size

# Simulate samples from the posterior Beta distribution
set.seed(123) # For reproducibility
posterior_samples <- rbeta(100000, alpha_posterior, beta_posterior)

# Point Estimators
posterior_mean <- mean(posterior_samples) # Posterior Mean
posterior_median <- median(posterior_samples) # Posterior Median

# HPD Interval
hpd_interval <- HPDinterval(as.mcmc(posterior_samples), prob = 0.95)

# Compute mode (if possible)
posterior_mode <- ifelse(alpha_posterior > 1 && beta_posterior > 1,
                         (alpha_posterior - 1) / (alpha_posterior + beta_posterior - 2),
                         NA) # Mode undefined for sharp or flat posterior

# Print the results
cat("Posterior Mean:", posterior_mean, "\n")
```

```
## Posterior Mean: 0.0005985882
```

```r
cat("Posterior Median:", posterior_median, "\n")
```

```
## Posterior Median: 0.0004651014
```

```r
cat("Posterior Mode (if valid):", posterior_mode, "\n")
```

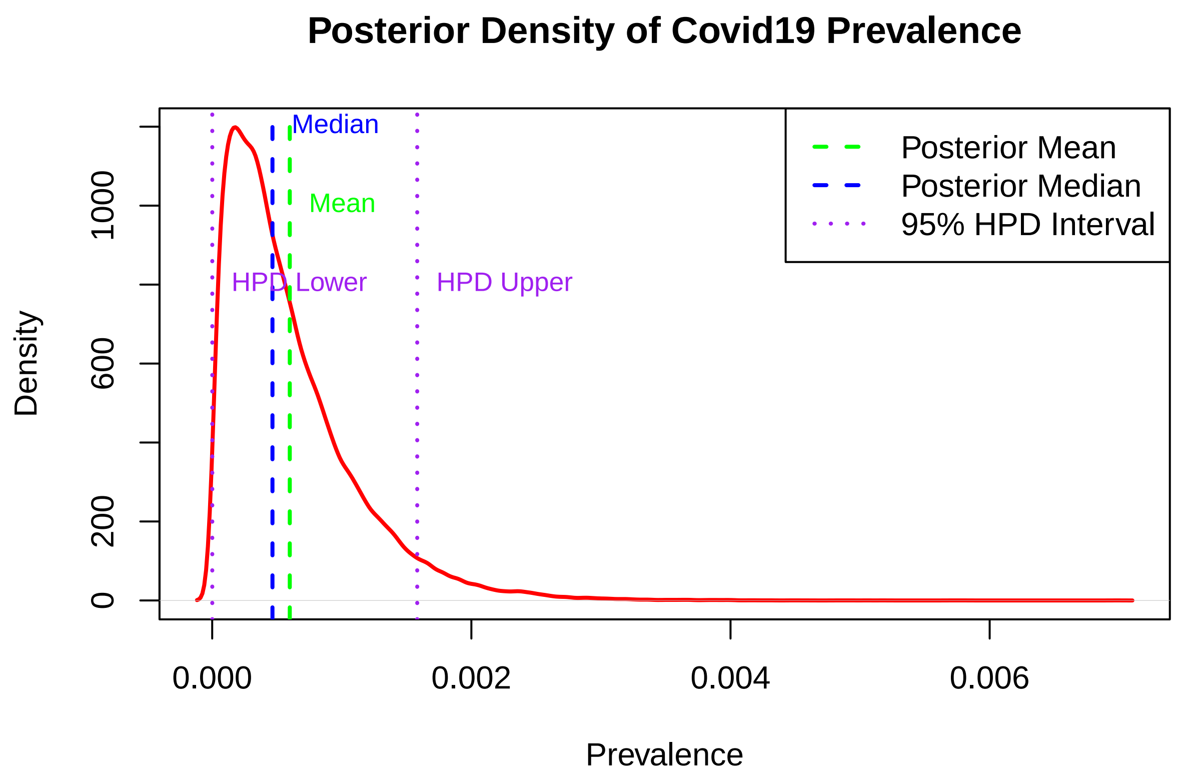
```
## Posterior Mode (if valid): 0.0001795407
```

```r
cat("95% HPD Interval:", hpd_interval[1], "-", hpd_interval[2], "\n")
```

```
## 95% HPD Interval: 5.620365e-07 - 0.001582044
```

```r
# Plot the posterior density
plot(density(posterior_samples),
     main = "Posterior Density of Covid19 Prevalence",
     xlab = "Prevalence",
     ylab = "Density",
     col = "red",
     lwd = 2)

# Add annotations for point estimates and HPD interval
abline(v = posterior_mean, col = "green", lty = 2, lwd = 2) # Mean
abline(v = posterior_median, col = "blue", lty = 2, lwd = 2) # Median
abline(v = hpd_interval[1], col = "purple", lty = 3, lwd = 2) # HPD lower bound
abline(v = hpd_interval[2], col = "purple", lty = 3, lwd = 2) # HPD upper bound

# Add a legend
```

```r
legend("topright",
       legend = c("Posterior Mean", "Posterior Median", "95% HPD Interval"),
       col = c("green", "blue", "purple"),
       lty = c(2, 2, 3),
       lwd = 2)

# Add text annotations
text(posterior_mean, 1000, "Mean", col = "green", pos = 4, cex = 0.8)
text(posterior_median, 1200, "Median", col = "blue", pos = 4, cex = 0.8)
text(hpd_interval[1], 800, "HPD Lower", col = "purple", pos = 4, cex = 0.8)
text(hpd_interval[2], 800, "HPD Upper", col = "purple", pos = 4, cex = 0.8)
```



Discussion

Statistik Austria chose Bayesian methods because they allow incorporating prior knowledge (e.g., German data) into the analysis, providing meaningful estimates even with zero positive cases in Austria. Unlike frequentist methods, which would yield a prevalence of zero or undefined intervals, Bayesian inference balances prior information with observed data. It also handles small sample sizes and extreme outcomes more effectively, producing interpretable intervals directly as probabilities.

Define Conjugate Priors for Bayesian Linear Modelling

In Bayesian linear modelling, we assume that the residuals are normally distributed. For a single explanatory variable scenario, the distribution of the response y is modeled as:

$$y \sim N(x^T\beta, \sigma^2)$$

Our goal is to define conjugate priors for the parameters: - **Coefficient (β)**, which represents the linear model's slope or intercept. - **Residual Variance (σ^2)**, which captures the variability of residuals.

Conjugate Priors 1. **Coefficient (β):** - The conjugate prior for β is a **Normal distribution**:

$$\beta \sim N(\mu_\beta, \tau^2)$$

- μ_β: Prior mean of the coefficient. - τ^2: Prior variance (or precision as $1/\tau^2$).

2. **Residual Variance (σ^2):**
   - The conjugate prior for σ^2 is an **Inverse-Gamma distribution**:

     $$\sigma^2 \sim \text{Inv-Gamma}(\alpha, \beta)$$

     - α: Shape parameter.
     - β: Scale parameter.

Setting Uninformative Priors Uninformative priors are used to minimize prior influence and let the data dominate the posterior distribution: - For β: Set $\mu_\beta = 0$ and $\tau^2 \to \infty$ (or a very large value). - For σ^2: Set $\alpha \to 0$ and $\beta \to 0$, or use a **Uniform prior** for the standard deviation σ, such as:

$$\sigma \sim U(0, L)$$

Comparison of Prior Parameters Different prior parameters influence the posterior differently: - **For β:** - Small τ^2 results in strong prior influence (informative prior). - Large τ^2 approximates uninformative priors. - **For σ^2:** - Large α, β favor smaller variances (informative). - Small α, β reduce prior influence (weakly informative).

Build the Normal Model for Regression Inference

we want to construct the corresponding normal model for Bayesian regression inference. We will derive the theoretical posterior distributions for the parameters β (coefficient) and σ^2 (residual variance) separately, under the assumption that the other parameter is "known."

Assumptions - The response variable y follows a normal distribution:

$$y \sim N(x^T\beta, \sigma^2)$$

- The prior distributions for the parameters are: - $\beta \sim N(\mu_\beta, \tau^2)$ - $\sigma^2 \sim \text{Inv-Gamma}(\alpha, \beta)$

Posterior Distributions

1. **Posterior for β (assuming σ^2 is known):**
   - The posterior distribution for β is also a normal distribution:

     $$\beta \mid y, \sigma^2 \sim N(\mu_\beta^*, \tau_\beta^2)$$

     where:

     $$\mu_\beta^* = \frac{\frac{\mu_\beta}{\tau^2} + \frac{x^T y}{\sigma^2}}{\frac{1}{\tau^2} + \frac{x^T x}{\sigma^2}}, \quad \tau_\beta^2 = \left(\frac{1}{\tau^2} + \frac{x^T x}{\sigma^2}\right)^{-1}$$

     - μ_β^*: Updated mean of β.
     - τ_β^2: Updated variance of β.
2. **Posterior for σ^2 (assuming β is known):**

- The posterior distribution for σ^2 is an inverse-gamma distribution:

$$\sigma^2 \mid y, \beta \sim \text{Inv-Gamma}\left(\alpha^*, \beta^*\right)$$

  where:

$$\alpha^* = \alpha + \frac{n}{2}, \quad \beta^* = \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2$$

  - α^*: Updated shape parameter.
  - β^*: Updated scale parameter.
  - n: Number of data points.

Point Estimators and 95% HPD Interval for Regression Parameters

we want to calculate the point estimators and 95% Highest Posterior Density (HPD) intervals for the regression parameters (β and σ^2), assuming the other parameter is "known."

1. **For β (assuming σ^2 is known):**

- **Posterior Distribution:**

$$\beta \mid y, \sigma^2 \sim N(\mu_\beta^*, \tau_\beta^2)$$

  where:

$$\mu_\beta^* = \frac{\frac{\mu_\beta}{\tau^2} + \frac{x^T y}{\sigma^2}}{\frac{1}{\tau^2} + \frac{x^T x}{\sigma^2}}, \quad \tau_\beta^2 = \left(\frac{1}{\tau^2} + \frac{x^T x}{\sigma^2}\right)^{-1}$$

- **Point Estimators:**
  - **Posterior Mean (Point Estimate):** μ_β^*
  - **Posterior Variance (Uncertainty):** τ_β^2

- **95% HPD Interval:** Assuming the posterior is normally distributed:

$$\text{HPD Interval for } \beta = \left[\mu_\beta^* - 1.96\sqrt{\tau_\beta^2}, \mu_\beta^* + 1.96\sqrt{\tau_\beta^2}\right]$$

---

2. **For σ^2 (assuming β is known):**

- **Posterior Distribution:**

$$\sigma^2 \mid y, \beta \sim \text{Inv-Gamma}(\alpha^*, \beta^*)$$

  where:

$$\alpha^* = \alpha + \frac{n}{2}, \quad \beta^* = \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2$$

- **Point Estimators:**
  - **Posterior Mean (Point Estimate):**

$$E[\sigma^2 \mid y, \beta] = \frac{\beta^*}{\alpha^* - 1} \quad (\text{for } \alpha^* > 1).$$

  - **Posterior Mode (Most Likely Value):**

$$\text{Mode}[\sigma^2 \mid y, \beta] = \frac{\beta^*}{\alpha^* + 1}$$

- **95% HPD Interval:** For the inverse-gamma distribution, numerical methods or specialized packages are typically used to compute the HPD interval. However, an approximate interval can be derived from the posterior distribution quantiles:

$$\text{HPD Interval for } \sigma^2 = [\text{Quantile}_{2.5\%}, \text{Quantile}_{97.5\%}]$$

Bayesian vs Frequentist Regression Analysis

we want to compare Bayesian regression results with frequentist results using the dataset `Auto` and the model:

$$\text{mpg} \sim \text{horsepower}.$$

```
# Load necessary libraries
library(MASS)     # For dataset
library(coda)     # For HPD intervals
library(ggplot2)  # For visualization
```

```
## Warning: Paket 'ggplot2' wurde unter R Version 4.3.3 erstellt
```

```
# Load the dataset
# load the data
data(Auto, package="ISLR")
attach(Auto)
```

```
## Das folgende Objekt ist maskiert package:ggplot2:
##
##     mpg
```

```
# Frequentist Regression: mpg ~ horsepower
freq_model <- lm(mpg ~ horsepower)
summary(freq_model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
# Extract frequentist estimates
freq_beta <- coef(freq_model)
freq_sigma2 <- summary(freq_model)$sigma^2
freq_confint <- confint(freq_model)

cat("Frequentist Results:\n")
```

```
## Frequentist Results:
```

```
cat("Beta (Intercept):", freq_beta[1], "\n")
```

```
## Beta (Intercept): 39.93586
```

```r
cat("Beta (Slope):", freq_beta[2], "\n")
```

```
## Beta (Slope): -0.1578447
```

```r
cat("Residual Variance (sigma^2):", freq_sigma2, "\n")
```

```
## Residual Variance (sigma^2): 24.06645
```

```r
cat("Confidence Intervals for Betas:\n", freq_confint, "\n")
```

```
## Confidence Intervals for Betas:
##  38.52521 -0.170517 41.34651 -0.1451725
```

```r
# Bayesian Regression
# Set prior parameters (uninformative priors)
alpha_prior <- 0.001
beta_prior <- 0.001
beta_0 <- 0     # Prior mean for beta
tau_0 <- 1e6    # Large variance (uninformative)

# Design matrix
X <- cbind(1, horsepower)
y <- mpg
n <- length(y)

# Posterior parameters for beta (assuming sigma^2 known)
sigma2_hat <- freq_sigma2  # Use frequentist residual variance as "known"
V_beta <- solve(t(X) %*% X / sigma2_hat + diag(1/tau_0, 2))  # Covariance
beta_hat <- V_beta %*% (t(X) %*% y / sigma2_hat + beta_0 / tau_0)

# Posterior distribution for sigma^2
residuals <- y - X %*% beta_hat
alpha_post <- alpha_prior + n / 2
beta_post <- beta_prior + sum(residuals^2) / 2

# Point estimates
beta_mean <- as.vector(beta_hat)
sigma2_mean <- beta_post / (alpha_post - 1)

# HPD intervals
beta_samples <- mvrnorm(10000, beta_mean, V_beta)  # Generate posterior samples for beta
beta_hpd <- apply(beta_samples, 2, function(col) HPDinterval(as.mcmc(col)))

# Posterior samples for sigma2
sigma2_samples <- 1 / rgamma(10000, alpha_post, rate = beta_post)
sigma2_hpd <- HPDinterval(as.mcmc(sigma2_samples))  # Wrap samples in mcmc before passing


cat("Bayesian Results:\n")
```

```
## Bayesian Results:
```

```r
cat("Beta (Intercept):", beta_mean[1], "\n")
```

```
## Beta (Intercept): 39.93584
```

```r
cat("Beta (Slope):", beta_mean[2], "\n")
```

```
## Beta (Slope): -0.1578446
```

```r
cat("Residual Variance (sigma^2):", sigma2_mean, "\n")
```

```
## Residual Variance (sigma^2): 24.06633
```

```r
cat("HPD Intervals for Betas:\n", beta_hpd, "\n")
```

```
## HPD Intervals for Betas:
##  38.53088 41.36734 -0.1702502 -0.1451739
```

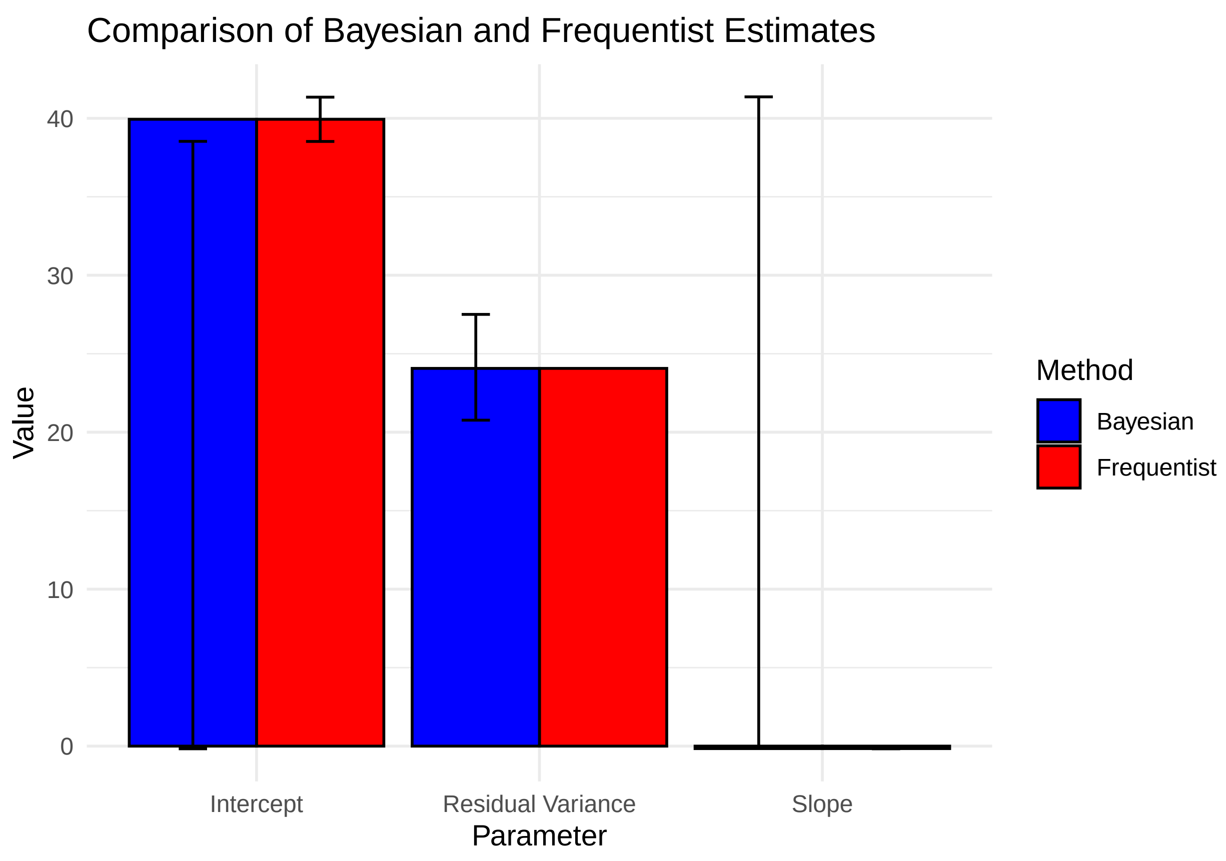
```r
cat("HPD Interval for sigma^2:\n", sigma2_hpd, "\n")
```

```
## HPD Interval for sigma^2:
##  20.766 27.50701
```

```r
library(ggplot2)
library(dplyr)
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.3.2 erstellt

##
## Attache Paket: 'dplyr'

## Das folgende Objekt ist maskiert 'package:MASS':
##
##     select

## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Prepare data for plotting
comparison_data <- data.frame(
  Parameter = rep(c("Intercept", "Slope", "Residual Variance"), each = 2),
  Method = rep(c("Frequentist", "Bayesian"), times = 3),
  Value = c(freq_beta[1], beta_mean[1], freq_beta[2], beta_mean[2], freq_sigma2, sigma2_mean),
  Lower = c(freq_confint[1, 1], beta_hpd[1, 1], freq_confint[2, 1], beta_hpd[2, 1], NA, sigma2_hpd[1]),
  Upper = c(freq_confint[1, 2], beta_hpd[1, 2], freq_confint[2, 2], beta_hpd[2, 2], NA, sigma2_hpd[2])
)


# Plot with error bars
ggplot(comparison_data, aes(x = Parameter, y = Value, fill = Method)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  geom_errorbar(aes(ymin = Lower, ymax = Upper), position = position_dodge(0.9), width = 0.2) +
  scale_fill_manual(values = c("blue", "red")) +
  labs(
    title = "Comparison of Bayesian and Frequentist Estimates",
    y = "Value",
    x = "Parameter"
  ) +
  theme_minimal()
```


Comparison of Bayesian and Frequentist Estimates
Value
40
30
20
10
0
Intercept
Residual Variance
Slope
Parameter
Method
Bayesian
Frequentist