Lab Report:Lab3-Group8

Udaya Shanker Mohanan Nair(udamo524), Uday Jain(udaja983)

2025-05-18

Assignment 1 : Gibbs samcol_sizeling for the logistic regression

```
##ASSIGNMENT 1

library(mvtnorm)
library(BayesLogit)

disease_data <- read.csv("Disease.csv")
x <- cbind(
  Intercept = 1,
  age_new = (disease_data$age -
               mean(disease_data$age))/sd(disease_data$age),
  gender = disease_data$gender,
  duration_new = (disease_data$duration_of_symptoms -
                    mean(disease_data$duration_of_symptoms))/sd(disease_data$duration_of_symptoms),
  dyspnoea = disease_data$dyspnoea,
  white_Blood_new = (disease_data$white_blood -
                       mean(disease_data$white_blood))/sd(disease_data$white_blood)
)

y <- disease_data$class_of_diagnosis
size <- nrow(x)
col_size <- ncol(x)

tau <- 3


gibbs_implementation <- function(x, y, n_iter=1000) {
  beta_samples <- matrix(0, n_iter, ncol(x))

  beta_value <- rep(0, ncol(x))
  size <- nrow(x)
  col_size <- ncol(x)

  prior_precision <- diag(1/tau^2, col_size)

  for (iter in 1:n_iter) {
    z <- x %*% beta_value
    poly_gamma <- rpg(size, 1, z)
    v <- solve(t(x) %*% diag(poly_gamma) %*% x + prior_precision)
```

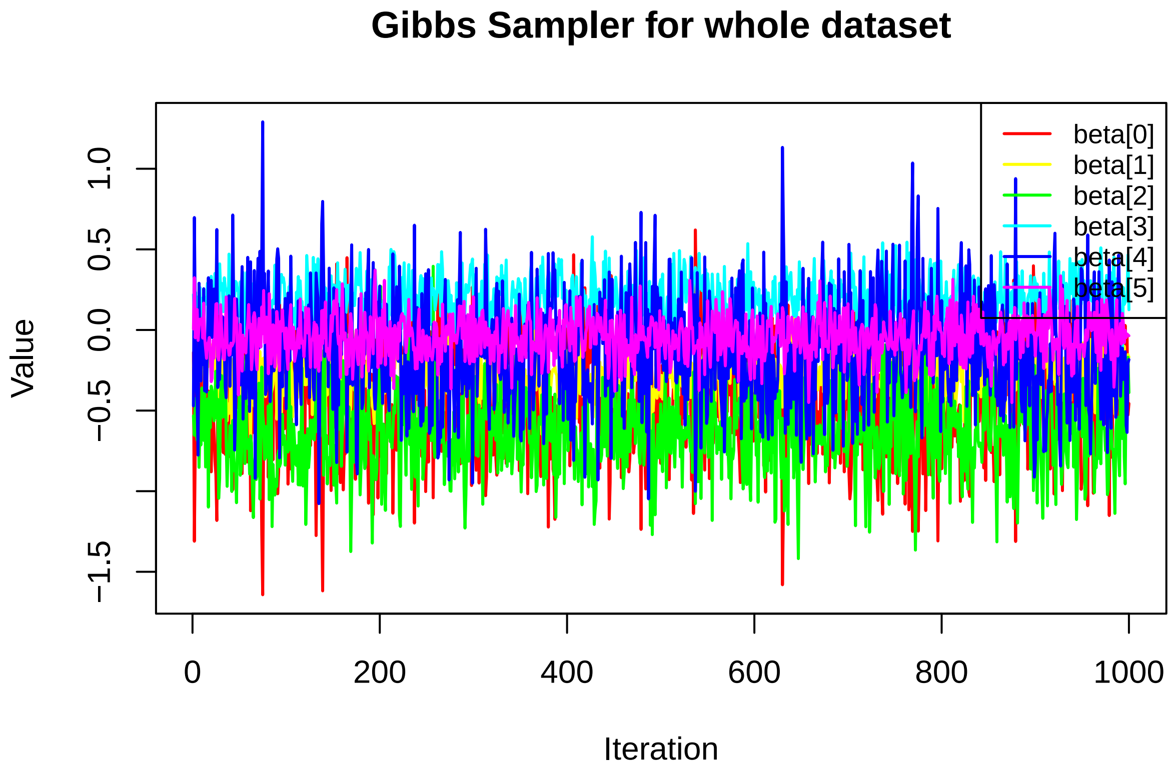
```r
    m <- v %*% t(x) %*% (y - 0.5)
    beta_value <- as.numeric(rmvnorm(1, m, v))
    beta_samples[iter,] <- beta_value
  }
  return(beta_samples)
}

set.seed(123)
results <- gibbs_implementation(x, y)

if_evaluation <- function(beta_values) {
  acf_value <- acf(beta_values, plot=FALSE, lag.max=50)$acf[,,1]
  if_value <- 1 + 2 * sum(acf_value[-1])
  return(if_value)
}

res_if <- apply(results, 2, if_evaluation)
cat("Inefficiency Factors for whole dataset:\n")
```

```
## Inefficiency Factors for whole dataset:
```

```r
print(res_if)
```

```
## [1] 1.7322117 0.9684822 0.9952840 0.6659212 2.2940092 0.4340299
```

```r
plot(1, type = "n",
     xlim = c(1, nrow(results)),
     ylim = range(results),
     xlab = "Iteration",
     ylab = "Value",
     main = "Gibbs Sampler for whole dataset")

cols <- rainbow(col_size)

for (i in 1:col_size) {
  lines(results[, i], col = cols[i], lwd = 1.5)
}
legend("topright",
       legend = paste0("beta[", 0:(col_size-1), "]"),
       col = cols,
       lty = 1,
       lwd = 1.5,
       cex = 0.8)
```



```
results_for_10_observations <- gibbs_implementation(x[1:10,], y[1:10])
results_for_40_observations <- gibbs_implementation(x[1:40,], y[1:40])
results_for_80_observations <- gibbs_implementation(x[1:80,], y[1:80])

res_if_10_observations <- apply(results_for_10_observations, 2, if_evaluation)
cat("Inefficiency Factors for 10 Observations:\n")
```

```
## Inefficiency Factors for 10 Observations:
```

```
print(res_if_10_observations)
```

```
## [1] 0.827587 1.420033 1.373011 2.115056 0.165961 2.412803
```

```
res_if_40_observations <- apply(results_for_40_observations, 2, if_evaluation)
cat("Inefficiency Factors for 40 Observations:\n")
```

```
## Inefficiency Factors for 40 Observations:
```

```
print(res_if_40_observations)
```

```
## [1] 1.8745666 1.0827795 1.9680445 0.2510922 1.7632683 2.3757253
```

```r
res_if_80_observations <- apply(results_for_80_observations, 2, if_evaluation)
cat("Inefficiency Factors for 80 Observations:\n")
```

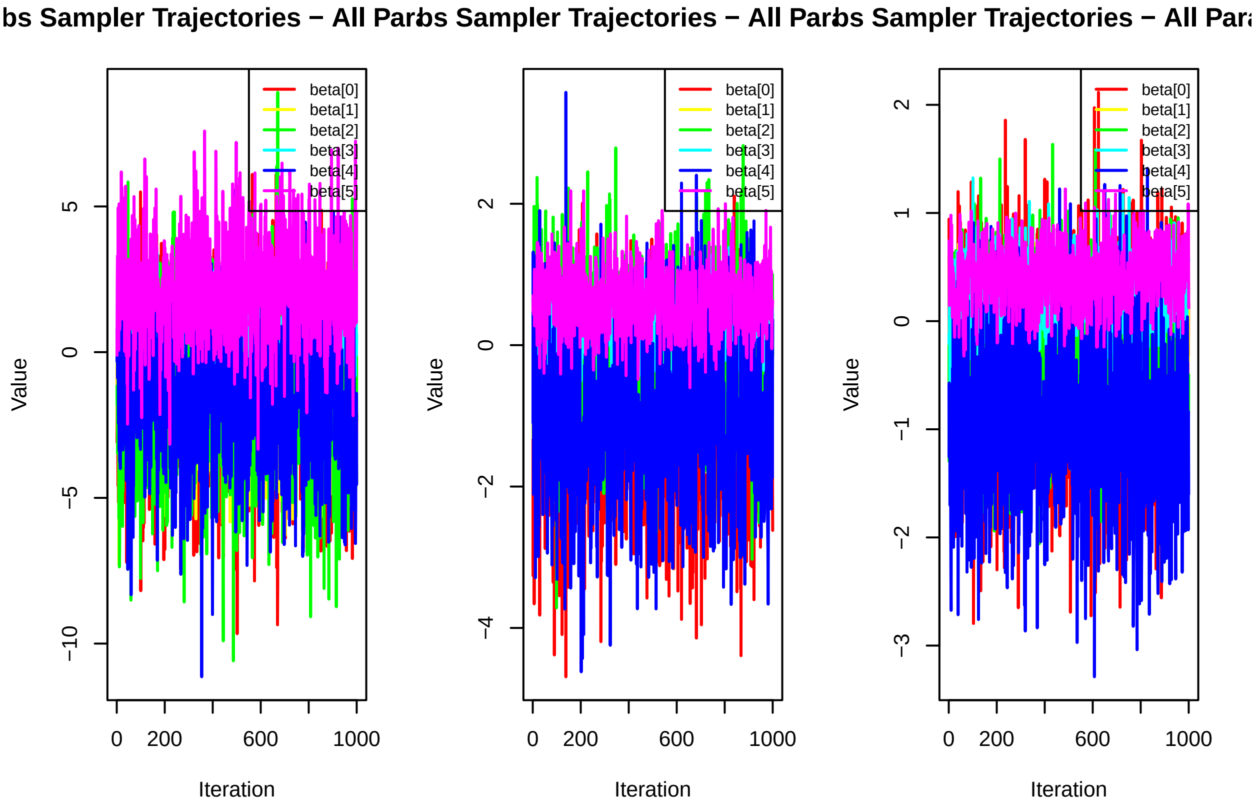
```
## Inefficiency Factors for 80 Observations:
```

```r
print(res_if_80_observations)
```

```
## [1] 1.3885638 1.0347889 1.3911514 0.9197756 1.6944566 3.0431779
```

```r
par(mfrow = c(1,3))
#10 Observations
plot(1, type = "n",
     xlim = c(1, nrow(results_for_10_observations)),
     ylim = range(results_for_10_observations),
     xlab = "Iteration",
     ylab = "Value",
     main = "Gibbs Sampler Trajectories - All Parameters")

cols <- rainbow(col_size)

for (i in 1:col_size) {
  lines(results_for_10_observations[, i], col = cols[i], lwd = 1.5)
}
legend("topright",
       legend = paste0("beta[", 0:(col_size-1), "]"),
       col = cols,
       lty = 1,
       lwd = 1.5,
       cex = 0.8)

#40 Observations

plot(1, type = "n",
     xlim = c(1, nrow(results_for_40_observations)),
     ylim = range(results_for_40_observations),
     xlab = "Iteration",
     ylab = "Value",
     main = "Gibbs Sampler Trajectories - All Parameters")

cols <- rainbow(col_size)

for (i in 1:col_size) {
  lines(results_for_40_observations[, i], col = cols[i], lwd = 1.5)
}
legend("topright",
       legend = paste0("beta[", 0:(col_size-1), "]"),
       col = cols,
       lty = 1,
       lwd = 1.5,
       cex = 0.8)

#80 Observations
```

```
plot(1, type = "n",
     xlim = c(1, nrow(results_for_80_observations)),
     ylim = range(results_for_80_observations),
     xlab = "Iteration",
     ylab = "Value",
     main = "Gibbs Sampler Trajectories - All Parameters")

cols <- rainbow(col_size)

for (i in 1:col_size) {
  lines(results_for_80_observations[, i], col = cols[i], lwd = 1.5)
}
legend("topright",
       legend = paste0("beta[", 0:(col_size-1), "]"),
       col = cols,
       lty = 1,
       lwd = 1.5,
       cex = 0.8)
```



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