MATH 8680: PROJECT 1 REPORT

Ali Salehi
Md Shiplu Hawlader
Instructor: Prof. E. Olusegun George
Bayesian Inference
June 11, 2018

Contents

1 Preprocessing of Data

The input data is:

	$Dose_i$	N_i	A_i
$i = 1$	1.69	60	6
$i = 2$	1.72	62	13
$i = 3$	1.76	63	20
$i = 4$	1.78	60	30
$i = 5$	1.81	64	53
$i = 6$	1.84	60	55
$i = 7$	1.86	62	61
$i = 8$	1.88	64	62

For each Binomial experiment i, we created N_i Bernoulli random variable y_{ij} in a way that A_i of them have value 1 and $(N_i - A_i)$ of them 0. Following is our final y vector:

```
print(y)
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[32] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0
[63] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[94] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
[125] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[156] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
[187] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
[218] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
[249] 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[280] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
[311] 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[342] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
[373] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[404] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1
[435] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[466] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

2 Problem Formulation

We are using same method introduced in [1] The fully conditional distribution for parameters β is:

$$\boldsymbol{\beta}|\boldsymbol{z}, \boldsymbol{X} \sim \mathcal{N}\Big((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{z}, (\boldsymbol{X}^T\boldsymbol{X})^{-1}\sigma^2\Big)$$

And the fully conditional for latent variables z is:

$$z_i|\boldsymbol{\beta}, y_i, \boldsymbol{x}_i \sim \begin{cases} \mathcal{TN}(\boldsymbol{x}_i\boldsymbol{\beta}, 1, 0, \infty) & \text{if } y_i = 1 \\ \mathcal{TN}(\boldsymbol{x}_i\boldsymbol{\beta}, 1, -\infty, 0) & \text{if } y_i = 0 \end{cases}$$

3 Results

3.1 Non Informative Prior $\tau(\beta) = 1$

	β_1	β_2	β_3
Mean Posterior	94.67	-125.43	40.60
MLE	83.87	-113.02	37.05

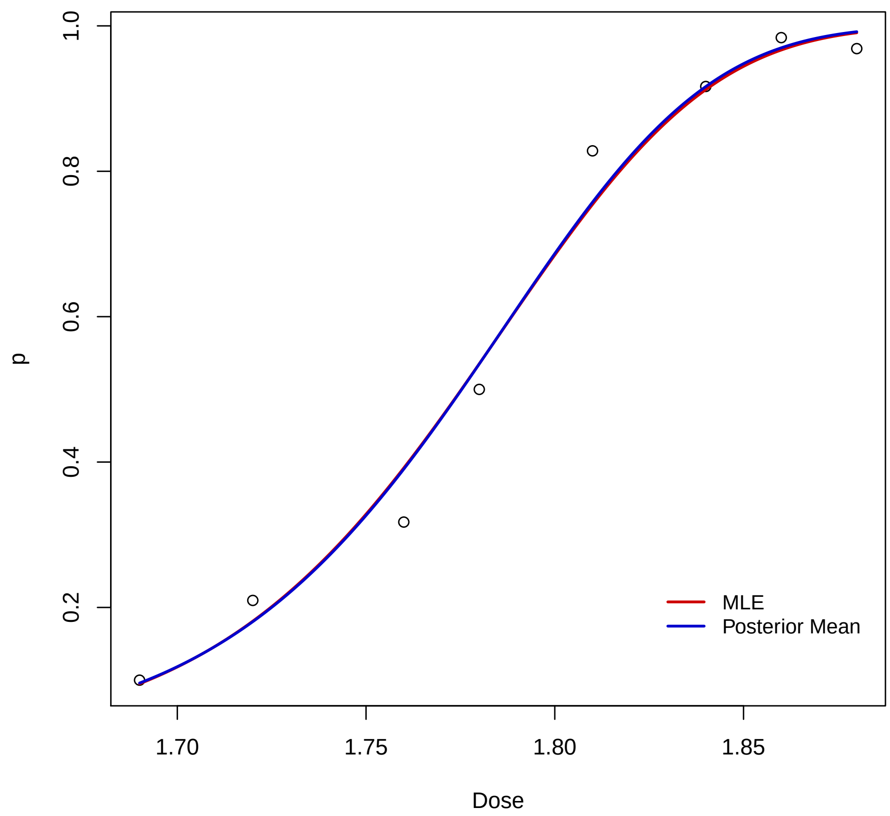

Figure 1: P(d) by estimated parameters (Non informative)

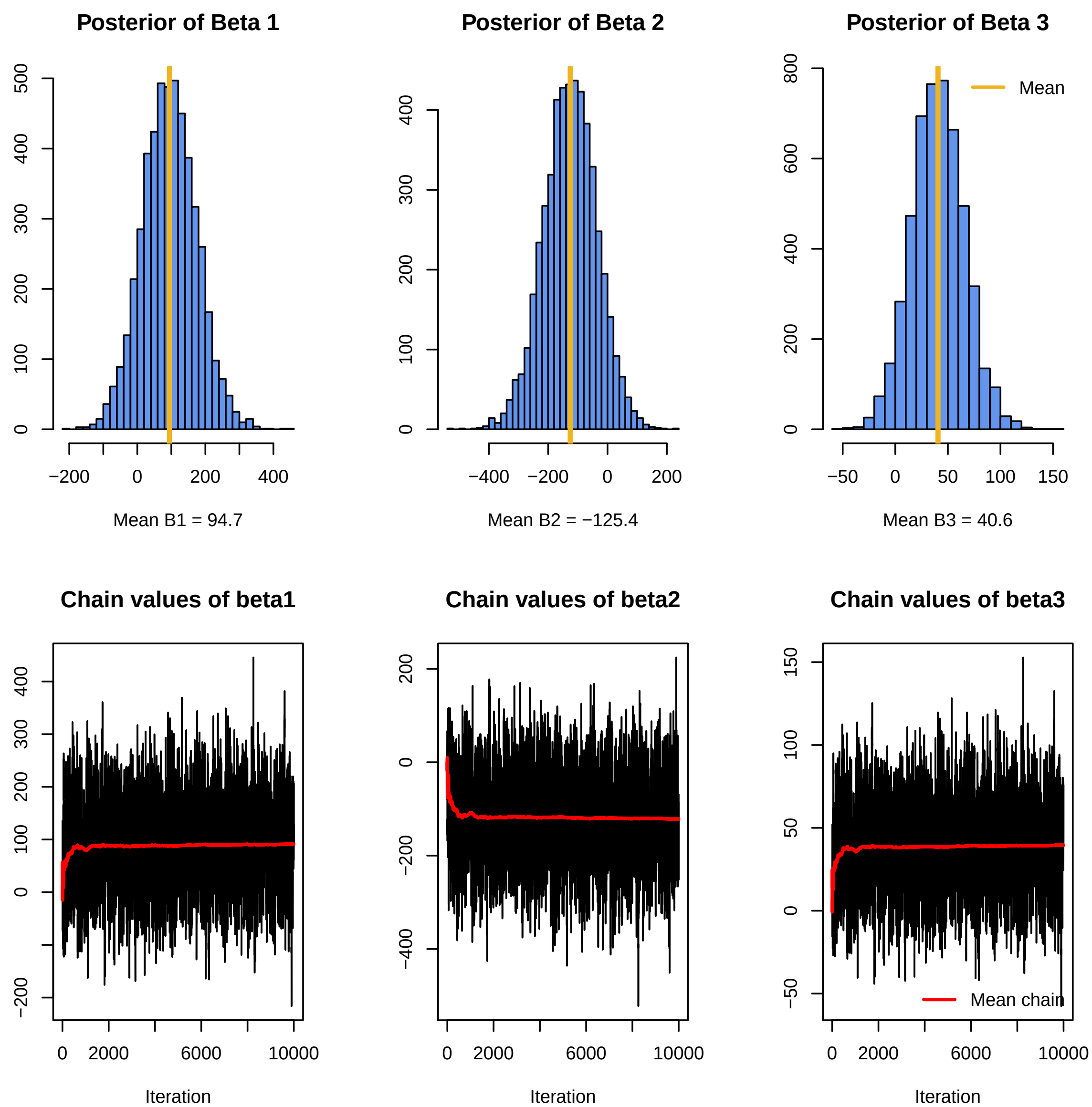

Figure 2: Histograms and Trace plots (Non informative)

3.2 Non Informative Prior $\tau(1/\sigma^2) \sim Gamma(1/2, 0.005)$

We decided to use three step Gibbs sampling to estimate β, z and σ^2 in the way presented in this paper. [2]. For this purpose, we need to have fully conditional distribution of $1/\sigma^2$, using the prior $\tau(1/\sigma^2) \sim Gamma(1/2, 0.005)$

For simplicity we put $\theta = \frac{1}{\sigma^2}$ and find the posterior of that:

$$\begin{aligned} f(y|\beta,\theta) &= \frac{1}{(2\pi)^2}\theta^{\frac{n}{2}} e^{\frac{-\theta}{2}(\underline{y}-X\underline{\beta})'(\underline{y}-X\underline{\beta})} \\ &= L(\beta,\theta|\underline{y}) \end{aligned}$$

Given prior

$$\tau(\theta) = Gamma(1/2, 0.005) = \frac{0.005^{\frac{1}{2}}}{\Gamma(\frac{1}{2})}\theta^{-\frac{1}{2}}e^{-0.005}$$

So

$$\begin{aligned}
\tau(\beta,\theta|\underline{y}) &\propto L(\beta,\theta)\tau(\theta) \\
&\propto \theta^{\frac{n}{2}}e^{-\frac{\theta}{2}(y-X\hat{\beta})'(y-X\hat{\beta})}e^{-\frac{\theta}{2}(\beta-\hat{\beta})'X'X(\beta-\hat{\beta})}\theta^{-\frac{1}{2}}e^{-0.005} \\
\tau(\theta|\beta,\underline{y},X) &\propto \theta^{\frac{n}{2}}e^{-\frac{\theta}{2}(\underline{y}-X\beta)'(\underline{y}-X\beta)}\theta^{-\frac{1}{2}}e^{-0.005} \\
&\propto \theta^{\frac{n-1}{2}}e^{-0.005\theta-\frac{\theta}{2}(\underline{y}-X\beta)'(\underline{y}-X\beta)} \\
&\propto \theta^{\frac{n-1}{2}}e^{-\theta(0.005+\frac{1}{2}(\underline{y}-X\beta)'(\underline{y}-X\beta)} \\
&\sim Gamma(\frac{n-1}{2}, 0.005+\frac{1}{2}(\underline{y}-X\beta)'(\underline{y}-X\beta))
\end{aligned}$$

Using this method the results are:

	β_1	β_2	β_3
Mean Posterior	91.34	-121.60	39.51
MLE	83.87	-113.02	37.05

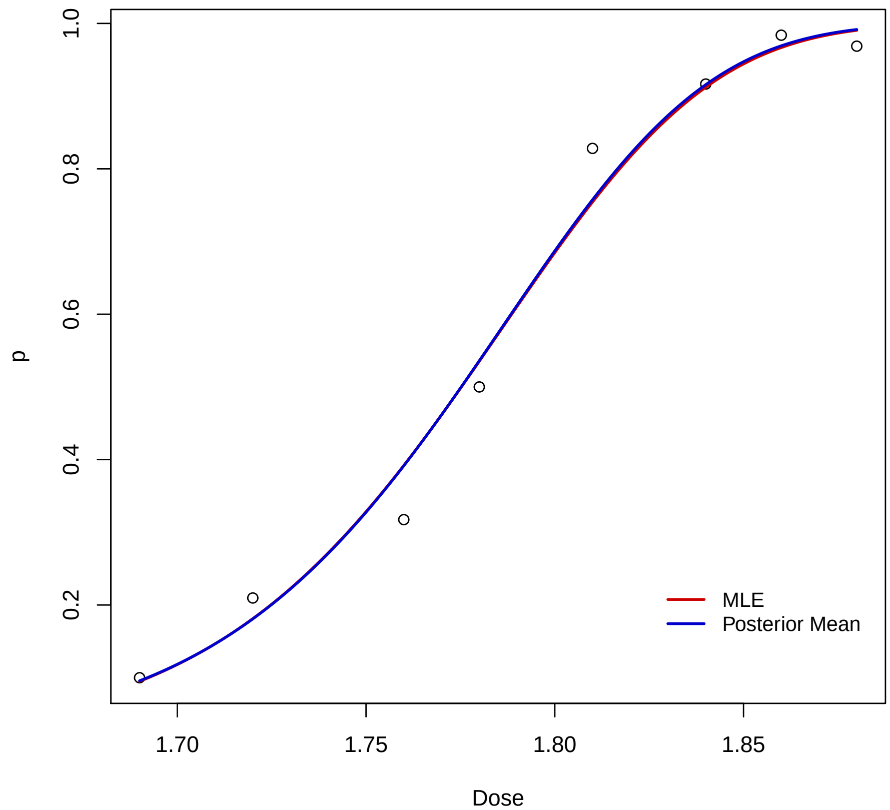

Figure 3: P(d) by estimated parameters (Non informative, $\tau(1/\sigma^2)$)

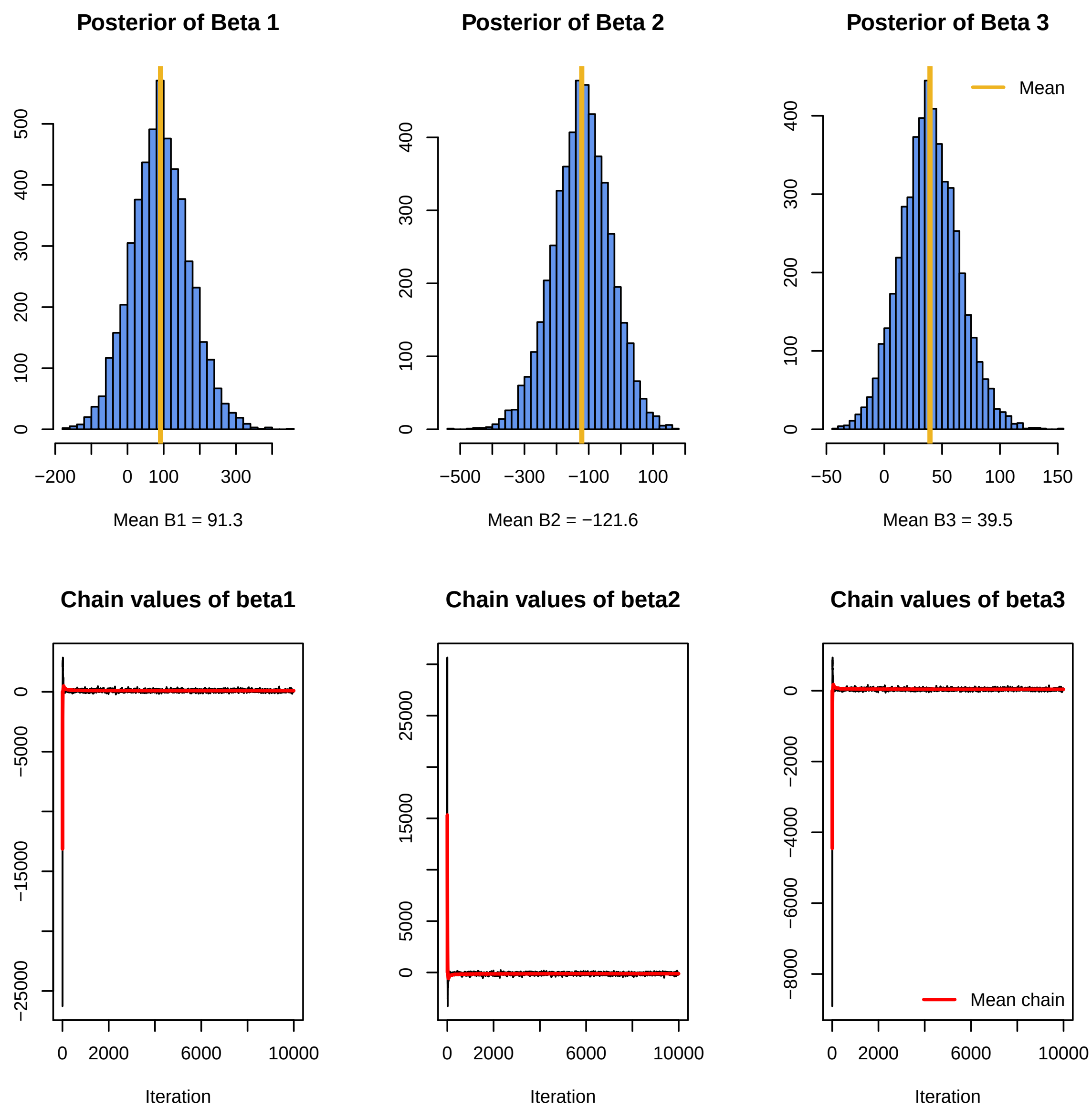

Figure 4: Histograms and Trace plots (Non informative, $\tau(1/\sigma^2)$)

3.3 Informative Prior $\tau(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{Q}_0)$

In which $\beta_0 = [000]$ and

$$\beta_0 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

and

$$Q_0 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$$

The fully conditional posterior will be:

$$\begin{aligned}\boldsymbol{\beta}|\boldsymbol{z},\boldsymbol{X} &\sim \mathcal{N}(\boldsymbol{M},\boldsymbol{V})\\ \boldsymbol{M} &= \boldsymbol{V}(\boldsymbol{Q}_0^{-1}\boldsymbol{\beta}_0 + \boldsymbol{X}^T\boldsymbol{z})\\ \boldsymbol{V} &= (\boldsymbol{Q}_0^{-1} + \boldsymbol{X}^T\boldsymbol{X})^{-1}\end{aligned}$$

[**?**] Using this prior the result are:

	β_1	β_2	β_3
Mean Posterior	-9.24	-6.45	6.59
MLE	83.87	-113.02	37.05

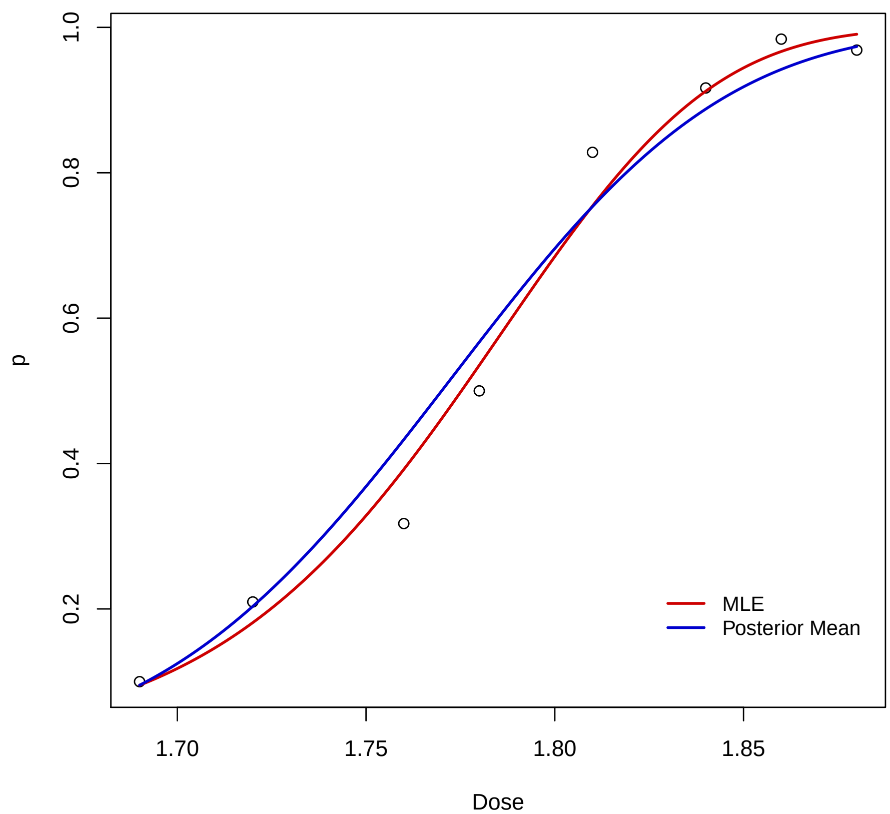

Figure 5: P(d) by estimated parameters (Informative)

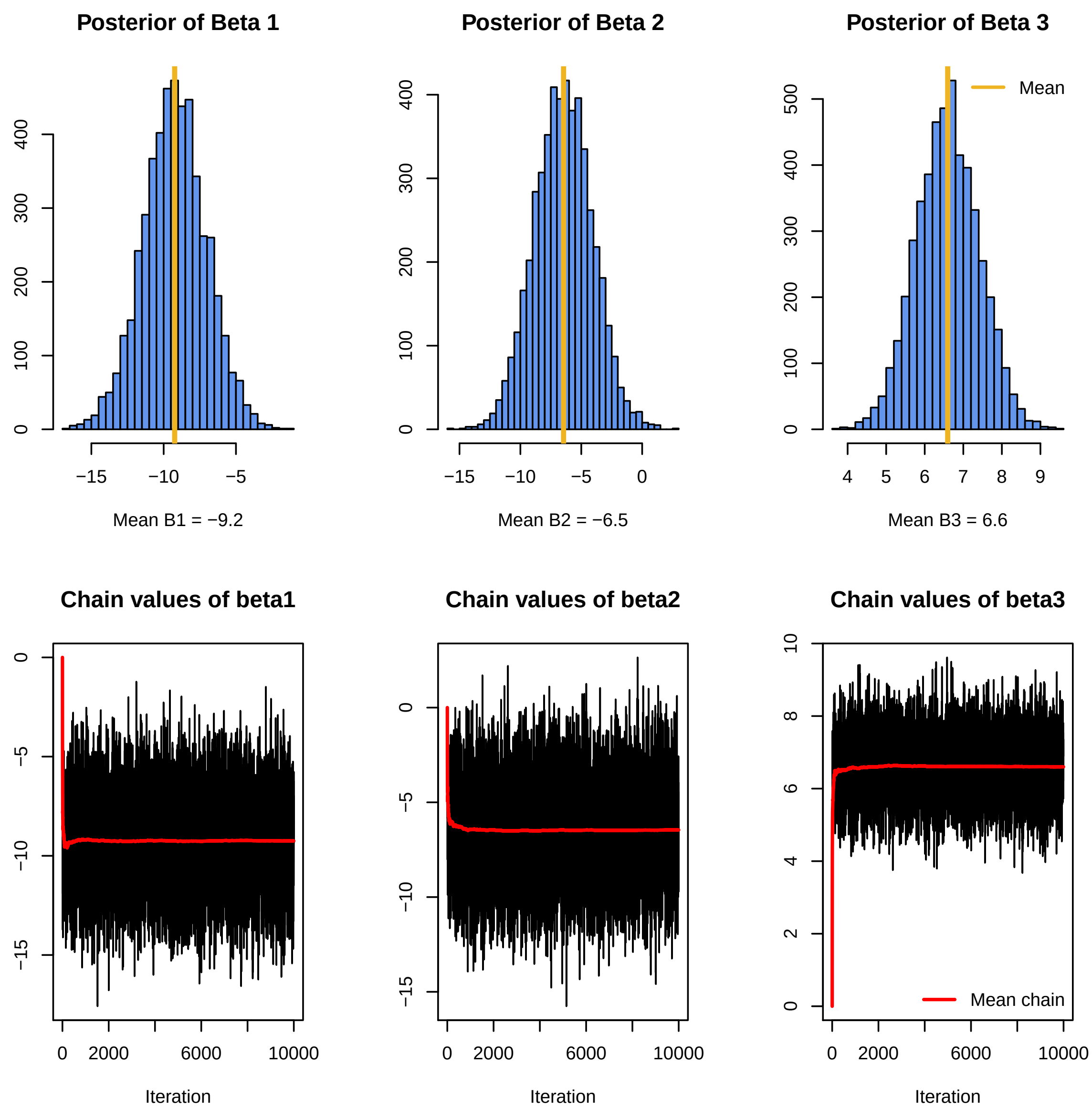

Figure 6: Histograms and Trace plots (Informative)

4 Implementation In R

```
#By:
# Ali Salehi
# Md Shiplu Hawlader


rm(list = ls()) # remove all variables from workspace
setwd("C:/Users/ali/Desktop/Google Drive/Courses/Bayesian Inference/project")
```

```
# Input data
Dose <- c(1.69, 1.72, 1.76, 1.78, 1.81, 1.84, 1.86, 1.88)
N <- c(60, 62, 63, 60, 64, 60, 62, 64)
A <- c(6, 13, 20, 30, 53, 55, 61, 62)
frequency <- A/N


# Convert to Bernoulli
d <- c() # empty vector
y <- c()
for (i in 1:length(Dose))
{
# create dose vector
tempD <- c(rep(Dose[i], N[i]))
d <- c(d , tempD)

# creatse y vector
temp0 <- c(rep(0, N[i]-A[i]))
temp1 <- c(rep(1, A[i]))
y <- c(y, temp0, temp1)
}

print(d)
print(y)

N <- length(y) # Number of samples
N1  <- sum(y)  # Number of successes
N0  <- N - N1  # Number of failures

# Form the X matrix
x <- cbind(d, d^2) # In this project we want to use d^2 also
D <- 3 # Number of predictors
X <- matrix(c(rep(1, length(d)), x), ncol = D) # X is complete data matrix

# MLE: fit the model to the data
fit <- glm(y ~ x , family = binomial(link = probit))
summary(fit)

mle_beta <- fit$coefficients


###################################################### Gibbs sampling

require(mvtnorm) # for sampling from Multivariate Normal distribution
require(truncnorm)# for sampling from Truncated Normal distribution

# Initialize parameters
```

```
beta <- rep(0, D)
z <- rep(0, N) # Latent variables

theta = 0.000001 # I use theta = 1/ (sigma^2)
prior_galpha <- 0.5 #0.1 # alpha of the prior distribution for (1/sigma^2 or thera)
prior_gbeta <- 0.005 #0.5 # beta of the prior distribution for (1/sigma^2 or thera)

# Gibbs sampler Parameters
N_sim <- 10000 # Number of simulations
burn_in <- 5000 # Burn in period
beta_chain <- matrix(0, nrow = N_sim, ncol = D) # Matrix storing samples of the \beta paramete

# Compute posterior variance of beta
X_trans_X_inv <- solve(crossprod(X, X)) #inverse of (X-transpose times X)
V <- X_trans_X_inv
X_trans_X_inv_X_trans <- tcrossprod (X_trans_X_inv, X) # Transpose of second parameter times t

post_galpha = N/2 - prior_galpha

method_flag = 2 # 0: non informative, 1: prior on sigma^2, 2: informative on beta
if (method_flag == 0){
print("Non informative!")
for (t in 2:N_sim)
{
# Update Mean of z
mu_z <- X %*% beta

# Draw latent variable z from its full conditional: z | beta, y, X, sigma^2
z[y == 0] <- rtruncnorm(N0, mean = mu_z[y == 0], sd = 1 , a = -Inf, b = 0)
z[y == 1] <- rtruncnorm(N1, mean = mu_z[y == 1], sd = 1 , a = 0, b = Inf)

# Update beta
M <- X_trans_X_inv_X_trans %*% z  # posterior mean of beta
V <- X_trans_X_inv # posterior covariance of beta
beta <- c(rmvnorm(1, M, V))# Draw variable beta from its full conditional: \beta | z, X, sigma

# Store the \beta draws
beta_chain[t, ] <- beta
}
curve_file <- "cr_nonInformative.pdf"
hist_file <- "hist_nonInformative.pdf"

} else if (method_flag == 1){
print("Prior on 1/Sima^2!")
for (t in 2:N_sim) {
# Update Mean of z
mu_z <- X %*% beta
```

```
# Draw latent variable z from its full conditional: z | beta, y, X, sigma^2
z[y == 0] <- rtruncnorm(N0, mean = mu_z[y == 0], sd = 1 , a = -Inf, b = 0)
z[y == 1] <- rtruncnorm(N1, mean = mu_z[y == 1], sd = 1 , a = 0, b = Inf)

# Update beta
M <- X_trans_X_inv_X_trans %*% z  # posterior mean of beta
V <- X_trans_X_inv * (1/theta) # posterior covariance of beta
beta <- c(rmvnorm(1, M, V))# Draw variable beta from its full conditional: \beta | z, X, sigma

# update theta (1 over sigma squared)
yxbeta <- y - (X %*% beta)
post_gbeta <- prior_gbeta + ((crossprod(yxbeta,yxbeta))/2)
theta <- rgamma(1, shape=post_galpha , rate = post_gbeta)
#print (1/theta)

# Store the \beta draws
beta_chain[t, ] <- beta
}
curve_file <- "cr_sigmaprior.pdf"
hist_file <- "hist_sigmaprior.pdf"

} else {
#informative prior on beta
print("Informative prior on beta!")
Q_0 <- diag(10, D)
prec_0 <- solve(Q_0)
beta_0 <- rep(0, D)
V <- solve(prec_0 + crossprod(X, X))
for (t in 2:N_sim) {
# Update Mean of z
mu_z <- X %*% beta
# Draw latent variable z from its full conditional: z | \beta, y, X
z[y == 0] <- rtruncnorm(N0, mean = mu_z[y == 0], sd = 1, a = -Inf, b = 0)
z[y == 1] <- rtruncnorm(N1, mean = mu_z[y == 1], sd = 1, a = 0, b = Inf)

# Compute posterior mean of theta
M <- V %*% (prec_0 %*% beta_0 + crossprod(X, z))
# Draw variable \beta from its full conditional: \theta | z, X
beta <- c(rmvnorm(1, M, V))

# Store the \beta draws
beta_chain[t, ] <- beta
}
curve_file <- "cr_informative.pdf"
hist_file <- "hist_informative.pdf"
}

# Get posterior mean of beta
```

```
post_beta <- colMeans(beta_chain[-(1:burn_in), ])


# Plot covariates x versus observations y

# Show the fitted function using the posterior mean estimates
xt <- seq(from = min(Dose), to = max(Dose ), by = 0.0001)
Xt <- matrix(c(rep(1, length(xt )), xt, xt^2), ncol = D)

dev.new()
pdf(file = curve_file)
plot(Dose, frequency, ylab="p")
lines(x = xt , y = pnorm(Xt %*% mle_beta), col = "red3", lwd = 2)
lines(x = xt, y = pnorm(Xt %*% post_beta), col = "blue3", lwd = 2)
legend("bottomright", legend=c("MLE","Posterior Mean"), col=c("red3","blue3"),
bty = 'n', lwd = 2, inset = c(0.02, 0.08), lty = 1, cex = 0.9)
dev.off()


# Plot histograms and Trace plots
dev.new()
pdf(file = hist_file)
par(mfrow = c(2,3))

hist(beta_chain[-(1:burn_in),1],breaks=30, main="Posterior of Beta 1",
xlab=paste('Mean B1',toString(round(post_beta[1],1)), sep=" = "),ylab="", col="cornflowerblue")
abline(v = post_beta[1], col="goldenrod2", lwd=3)

hist(beta_chain[-(1:burn_in),2], breaks=30, main="Posterior of Beta 2",
xlab=paste('Mean B2',toString(round(post_beta[2],1)), sep=" = "),ylab="", col="cornflowerblue")
abline(v = post_beta[2], col="goldenrod2", lwd=3)

hist(beta_chain[-(1:burn_in),3], breaks=30, main="Posterior of Beta 3",
xlab=paste('Mean B3',toString(round(post_beta[3],1)), sep=" = "),ylab="", col="cornflowerblue")
abline(v = post_beta[3], col="goldenrod2", lwd=3)

legend("topright", c("Mean"), lty=1, lwd=2,
col=c("goldenrod2"), bty='n', cex=.95)


plot(beta_chain[, 1], type = "l", xlab="Iteration" , ylab="",
main = "Chain values of beta1")
lines(cumsum(beta_chain[, 1])/(1:N_sim), col="red", lwd=2)

plot(beta_chain[, 2], type = "l", xlab="Iteration" , ylab="",
main = "Chain values of beta2")
lines(cumsum(beta_chain[, 2])/(1:N_sim), col="red", lwd=2)
```

```
plot(beta_chain[, 3], type = "l", xlab="Iteration" , ylab="",
main = "Chain values of beta3")
lines(cumsum(beta_chain[, 3])/(1:N_sim), col="red", lwd=2)
legend("bottomright", c("Mean chain"), lty=1, lwd=2,
col=c("red"), bty='n', cex=.95)

dev.off()

print (matrix(round(mu_z,2), nrow = 10, byrow = FALSE))
print (post_beta)
print (mle_beta)
```

References

[1] J. H. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *Journal of the American statistical Association*, vol. 88, no. 422, pp. 669–679, 1993.

[2] C. Czado, "Bayesian inference of binary regression models with parametric link," *Journal of Statistical Planning and Inference*, vol. 41, no. 2, pp. 121–140, 1994.