

Homeworks 9 and 10

STT 465, Bayesian Statistical Methods

Lowell Monis

December 8, 2025

Question 1

The file “msparrownest.txt” contains data on the nesting success of 43 young male sparrows of the same age, as well as their wingspan. Download the file onto your computer, import the data into R using:

```
nest <- read.table("data/msparrownest.txt"); y <- nest[, 1]; wspan <- nest[, 2]
```

Then, y is the indicator of nesting success (“1” denotes success and “0” denotes failure), and $wspan$ is the wingspan. Let Y_i be the binary indicator that sparrow i successfully nests, and let x_i denote its wingspan. We model Y_i ’s as independent Bernoulli distribution with the following success probability.

$$P(Y_i = 1 \mid x_i, \beta_1, \beta_2) = \frac{e^{\beta_1 + \beta_2 x_i}}{1 + e^{\beta_1 + \beta_2 x_i}}, \quad i = 1, 2, \dots, 43$$

We denote

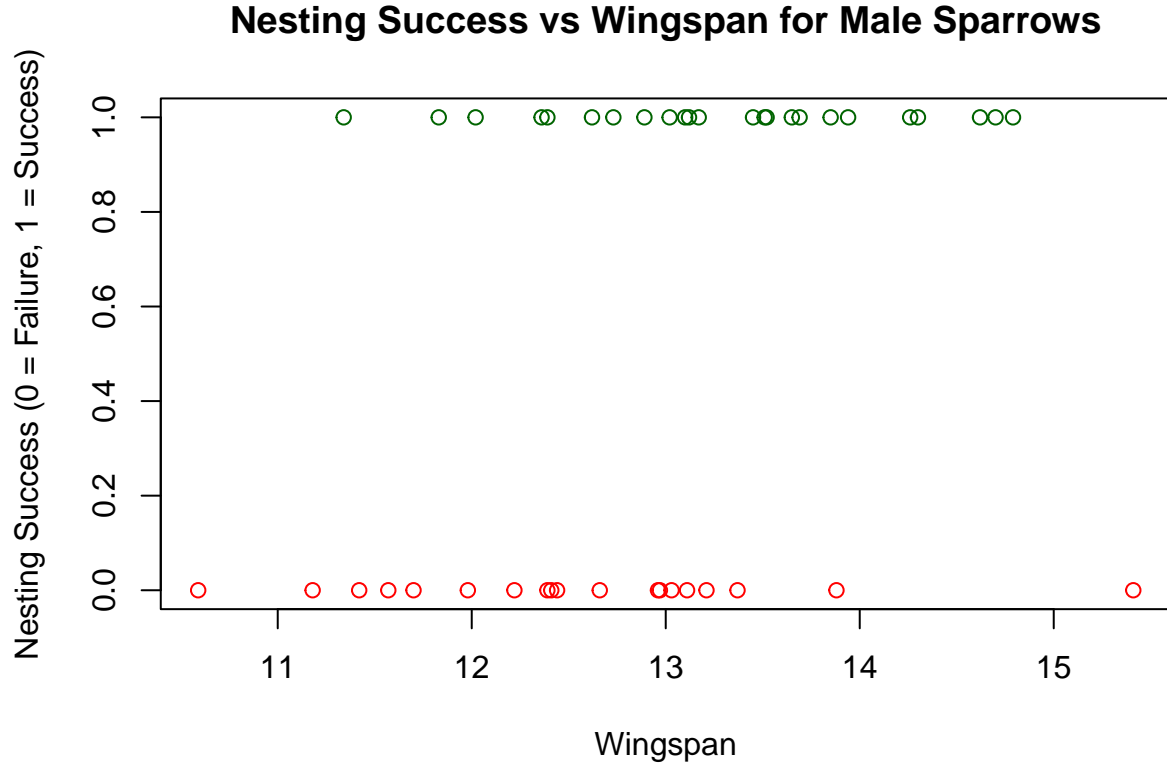
$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{43} \end{pmatrix}$$

(a) 30 points

Make a scatter plot between y and $wspan$. What do you observe regarding the relationship between the nesting success and wingspan? Will the model above be able to capture such a relationship?

Answer

```
plot(wspan, y,
     xlab = "Wingspan",
     ylab = "Nesting Success (0 = Failure, 1 = Success)",
     main = "Nesting Success vs Wingspan for Male Sparrows",
     col = ifelse(y == 1, "darkgreen", "red"))
```



There is a visible positive association between wingspan and nesting success. Sparrows with larger wingspans are more likely to successfully nest.

From the above plot, there appears to be two distinct classes—two distinct clusters that are very clear to indicate a logistic model due to the potential binary outcome being almost perfect. 1 denotes nesting success, while 0 denotes nesting failure. I am able to visualize a logistic function on the data such that it uses the model above. This is because I see a near S-shaped plot taking shape. The S-shaped logistic curve appropriately models how probability of success increases with wingspan. The monotonic increasing trend in the data matches the logistic model structure. There are, however, potential outliers, with a lot the data clusters overlapping when I draw a vertical line on the plot. So while this can be modeled by a logistic curve, its accuracy would not be very high. I can expect an S-shaped plot, since the model is indeed a logistic function, and the data looks friendly to a logistic model on this plot.

(b) 30 points

We consider the multivariate normal prior for β : $p(\beta) = \mathcal{N}(\mu_0, \Sigma_0)$ where $\mu_0 = (-3.35, 0.2)$, $\Sigma_0 = \begin{pmatrix} 200 & 0 \\ 0 & 100 \end{pmatrix}$. Equivalently speaking, we consider independent prior of β_1, β_2 with $\beta_1 \sim \mathcal{N}(-3.35, 200)$ and $\beta_2 \sim \mathcal{N}(0.2, 100)$. We then use the proposal distribution $J(\beta^* | \beta^{(k)}) = \mathcal{N}(\beta^{(k)}, \Sigma)$ with $\Sigma = \sigma^2(\mathbf{X}^T \mathbf{X})^{-1}$. Run the Metropolis algorithm with $\sigma^2 \in \{0.6, 6, 60\}$ for 1000 iterations. Compute the acceptance rate and lag 1-3 autocorrelations for each value of σ^2 . Which value would you pick for the use of σ^2 ?

Answer

We model the nesting success Y_i for sparrow i as an independent Bernoulli random variable with success probability:

$$P(Y_i = 1 | \pi_i, \beta_1, \beta_2) = \frac{e^{\beta_1 + \beta_2 x_i}}{1 + e^{\beta_1 + \beta_2 x_i}}, \quad i = 1, 2, \dots, 43$$

where x_i is the wingspan of sparrow i .

We use a multivariate normal prior for $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$:

$$p(\beta) = \mathcal{N}(\mu_0, \Sigma_0)$$

$$\text{where } \mu_0 = \begin{pmatrix} -3.35 \\ 0.2 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 200 & 0 \\ 0 & 100 \end{pmatrix}.$$

This is equivalent to independent priors β_1, β_2 with $\beta_1 \sim \mathcal{N}(-3.35, 200)$ and $\beta_2 \sim \mathcal{N}(0.2, 100)$.

The question asks us to use the proposal distribution:

$$J(\beta^* | \beta^{(k)}) = \mathcal{N}(\beta^{(k)}, \sigma^2(\mathbf{X}^T \mathbf{X})^{-1})$$

where $\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{43} \end{pmatrix}$ is the design matrix.

The acceptance ratio on the log scale is:

$$\log r = \sum_{i=1}^{43} \left[\log p(y_i | \beta^*) - \log p(y_i | \beta^{(k)}) \right] + \left[\log p(\beta^*) - \log p(\beta^{(k)}) \right]$$

where the Bernoulli log-likelihood is:

$$\log p(y_i | \beta) = y_i \log(\pi_i(\beta)) + (1 - y_i) \log(1 - \pi_i(\beta))$$

$$\text{with } \pi_i(\beta) = \frac{e^{\beta_1 + \beta_2 x_i}}{1 + e^{\beta_1 + \beta_2 x_i}}$$

The Metropolis algorithm will be implemented as follows:

First, we initialize a starting value $\beta^{(1)}$

Then, for each iteration $k = 1, 2, \dots, S$:

- Sample proposal $\beta^* \sim \mathcal{N}(\beta^{(k)}, \sigma^2(\mathbf{X}^T \mathbf{X})^{-1})$
- Compute $\log r$
- Sample $u \sim \text{Uniform}(0, 1)$
- Set $\beta^{(k+1)} = \begin{cases} \beta^*, & \text{if } \log u \leq \log r \text{ (accept proposal)} \\ \beta^{(k)}, & \text{else (reject proposal)} \end{cases}$

We will be conducting diagnostics with the acceptance rate of the proposals, and the measure dependence between samples at different lags using the lag 1-3 autocorrelation. Lower autocorrelation indicates better mixing and more efficient sampling.

We will need to import the `mvtnorm` package to proceed with the algorithm.

```
# imports, data, and constants
library(mvtnorm)
X <- cbind(rep(1, length(y)), wspan) # design matrix
n <- length(y); p <- dim(X)[2]
mu0 <- c(-3.35, 0.2); Sigma0 <- matrix(c(200,0,0,100), nrow=2, byrow=TRUE) # prior
XtX_inv <- solve(t(X)%*%X)
```

```

S <- 1000 # iterations
sigma2_vals <- c(0.6,6,60)
results <- list()
set.seed(465)

for (i in 1:length(sigma2_vals)) {
  sigma2 <- sigma2_vals[i]

  # storage
  BETA <- matrix(0, nrow=S, ncol=p)
  BETA[1, ] <- c(0,0)
  beta <- c(0, 0) # reset
  acp <- 0

  Sigma_prop <- sigma2*XtX_inv
  for (k in 2:S) {
    beta.p <- as.vector(rmvnorm(1, mean=beta, sigma=Sigma_prop))
    pi.p <- exp(X %*% beta.p) / (1 + exp(X %*% beta.p))
    pi <- exp(X %*% beta) / (1 + exp(X %*% beta))
    log.r <- sum(dbinom(y, prob = pi.p, size = 1, log = T)) -
      sum(dbinom(y, prob = pi, size = 1, log = T)) +
      dmvnorm(beta.p, mu0, Sigma0, log = T) -
      dmvnorm(beta, mu0, Sigma0, log = T)

    if(log(runif(1))<=log.r){
      beta <- beta.p
      acp <- acp + 1
    }

    BETA[k, ] <- beta
  }
  acf_beta1 <- acf(BETA[, 1], lag.max = 3, plot = FALSE)
  acf_beta2 <- acf(BETA[, 2], lag.max = 3, plot = FALSE)
  b1_lag1 <- acf_beta1$acf[2]
  b1_lag2 <- acf_beta1$acf[3]
  b1_lag3 <- acf_beta1$acf[4]
  b2_lag1 <- acf_beta2$acf[2]
  b2_lag2 <- acf_beta2$acf[3]
  b2_lag3 <- acf_beta2$acf[4]
  acceptance_rate = acp/S
  cat(sprintf("sigma^2 = %.1f: Acceptance rate = %.3f,\n
    Beta_1: lag-1 is %.3f, lag-2 is %.3f, lag-3 is %.3f\n
    Beta_2: lag-1 is %.3f, lag-2 is %.3f, lag-3 is %.3f\n\n",
    sigma2, acceptance_rate, b1_lag1, b1_lag2, b1_lag3,
    b2_lag1, b2_lag2, b2_lag3))
}

```

```

## sigma^2 = 0.6: Acceptance rate = 0.850,
##
##           Beta_1: lag-1 is 0.950, lag-2 is 0.905, lag-3 is 0.860
##
##           Beta_2: lag-1 is 0.950, lag-2 is 0.904, lag-3 is 0.858
##

```

```
## sigma^2 = 6.0: Acceptance rate = 0.531,
##
##           Beta_1: lag-1 is 0.816, lag-2 is 0.662, lag-3 is 0.546
##
##           Beta_2: lag-1 is 0.823, lag-2 is 0.674, lag-3 is 0.561
##
## sigma^2 = 60.0: Acceptance rate = 0.159,
##
##           Beta_1: lag-1 is 0.826, lag-2 is 0.690, lag-3 is 0.562
##
##           Beta_2: lag-1 is 0.825, lag-2 is 0.687, lag-3 is 0.556
```

With the above computations conducted, it looks like the optimal value is $\sigma^2 = 6.0$, since the acceptance rate of 53.1% is closest to the optimal range of 20-50%. It fits the criteria that the proposal variance is large enough such that the Markov chain can move around the parameter space, with the first three autocorrelation lags going from 0.816 to 0.662 to 0.546, but not so large that the proposals end up getting rejected most of the time. The candidate value of 60.0 also fits the autocorrelation requirements. The rate of convergence is increased with the decreased correlation in the chain for both these values. However, the acceptance rate for 60.0 is very low.

(c) 30 points

Using the value of σ^2 you picked from (1b), re-run the Metropolis algorithm for 20,000 iterations. Perform the usual MCMC diagnostics and then compute the posterior mean and 95% credible interval for β_1 and β_2 . Furthermore, compute the maximum likelihood estimates of β_1 and β_2 and compare the results.

Answer

I have selected $\sigma^2 = 6.0$. I will co-opt the code from (1b), modifying it to fit the requirements of this question. We will create 20,000 iterations this time and follow the same algorithm.

```
S <- 20000 # iterations
results <- list()
set.seed(465)
sigma2 <- 6.0

BETA <- matrix(0, nrow=S, ncol=p)
BETA[1, ] <- c(0,0)
acp <- 0

for (k in 2:S) {
  beta.p <- as.vector(rmvnorm(1, mean=beta, sigma=Sigma_prop))
  pi.p <- exp(X %*% beta.p) / (1 + exp(X %*% beta.p))
  pi <- exp(X %*% beta) / (1 + exp(X %*% beta))
  log.r <- sum(dbinom(y, prob = pi.p, size = 1, log = T)) -
    sum(dbinom(y, prob = pi, size = 1, log = T)) +
    dmvnorm(beta.p, mu0, Sigma0, log = T) -
    dmvnorm(beta, mu0, Sigma0, log = T)

  if(log(runif(1))<=log.r){
    beta <- beta.p
    acp <- acp + 1
  }
}
```

```

}

BETA[k, ] <- beta
}

```

We can now proceed with diagnostics.

```

par(mfrow = c(2, 2))

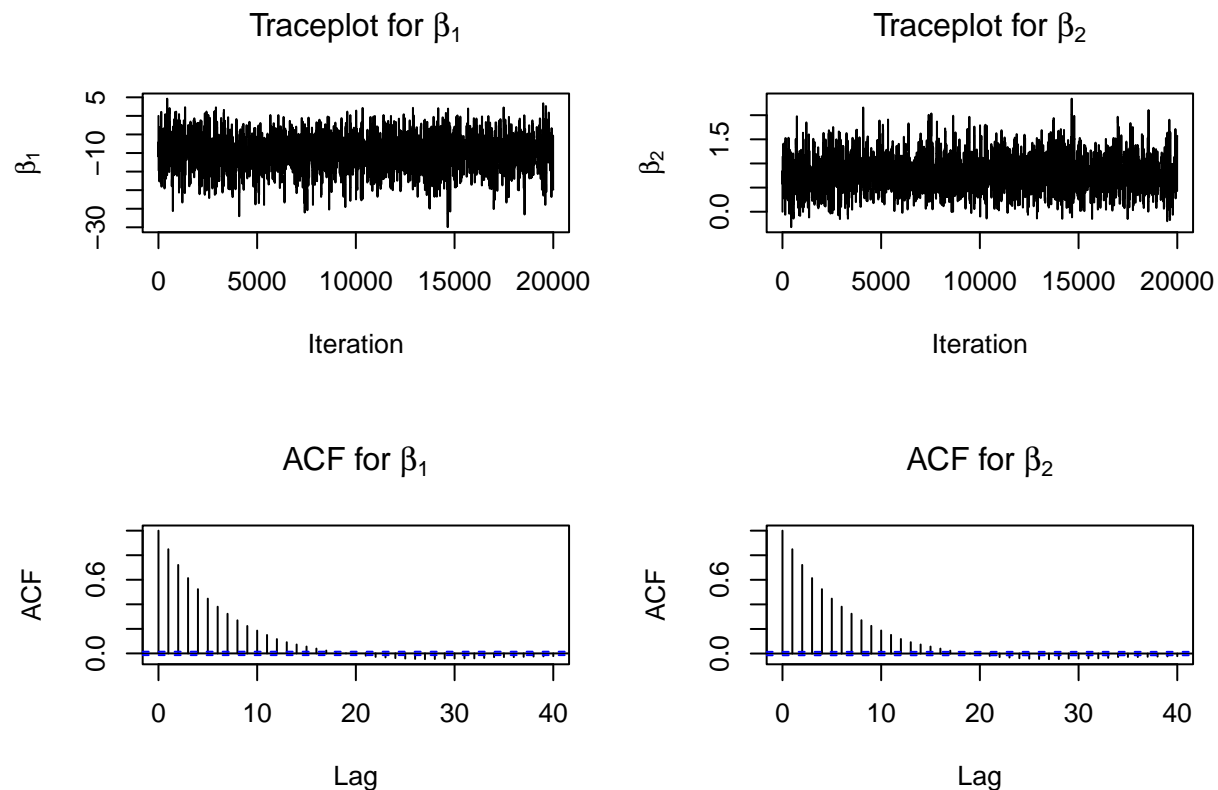
plot(1:S, BETA[, 1], type = "l", xlab = "Iteration", ylab = expression(beta[1]),
     main = expression(paste("Traceplot for ", beta[1])))

plot(1:S, BETA[, 2], type = "l", xlab = "Iteration", ylab = expression(beta[2]),
     main = expression(paste("Traceplot for ", beta[2])))

acf(BETA[, 1], main = expression(paste("ACF for ", beta[1])),
    lag.max = 40)

acf(BETA[, 2], main = expression(paste("ACF for ", beta[2])),
    lag.max = 40)

```



The traceplots demonstrate that there is good mixing in both β_1 and β_2 . The chains are exploring the predictor space well. We know this since the plots are like fuzzy caterpillars. We can also conclude from this that the autocorrelation is not high. There is almost no burn-in period, and rapid movement across a stable range of values and a stable mean, indicating the chain has converged to the posterior distribution. From the autocorrelation plots, we can see that the autocorrelation is indeed low, with decay to zero happening within 15 lags for both parameters. The moderately fast decay shows that the data is explored pretty well by the chain, and the chain converges quickly.

Next, we compute the posterior means for both parameters, as well as their 95% credible intervals.

```
post_mean_beta1 <- mean(BETA[, 1])
post_mean_beta2 <- mean(BETA[, 2])
ci_beta1 <- quantile(BETA[, 1], probs = c(0.025, 0.975))
ci_beta2 <- quantile(BETA[, 2], probs = c(0.025, 0.975))
```

The posterior mean for β_1 is `rpost_mean_beta1`, which lies in its 95% credible interval [-19.1934009723226, -1.35621161047069].

The posterior mean for β_2 is `rpost_mean_beta2`, which lies in its 95% credible interval [0.119713935863257, 1.50505164852999].

Next, we compute the Frequentist/classical maximum likelihood estimates for the parameters.

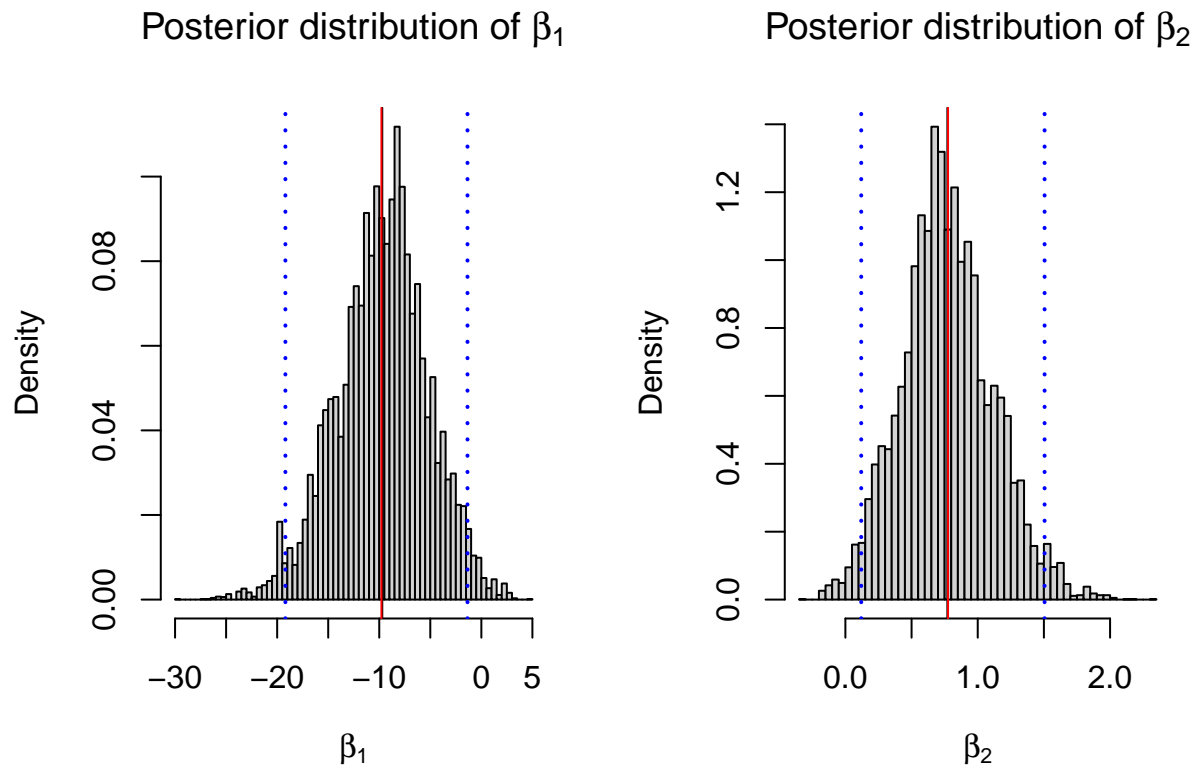
```
fit.mle <- glm(y ~ wspan, family = "binomial")
mle_beta1 <- fit.mle$coeff[1]
mle_beta2 <- fit.mle$coeff[2]
```

For β_1 , the MLE is -9.78681, and for β_2 it is 0.7757703. We can now visualize this information on a posterior distribution plot. The blue lines denote the 95% credible intervals, the black lines denote the posterior mean, and the red lines denote the classical MLE.

```
par(mfrow = c(1, 2))

hist(BETA[, 1], breaks = 50, probability = TRUE,
     xlab = expression(beta[1]),
     main = expression(paste("Posterior distribution of ", beta[1])))
abline(v = post_mean_beta1, col = "black", lwd = 1, lty = 1)
abline(v = mle_beta1, col = "red", lwd = 1, lty = 1)
abline(v = ci_beta1, col = "blue", lwd = 2, lty = 3)

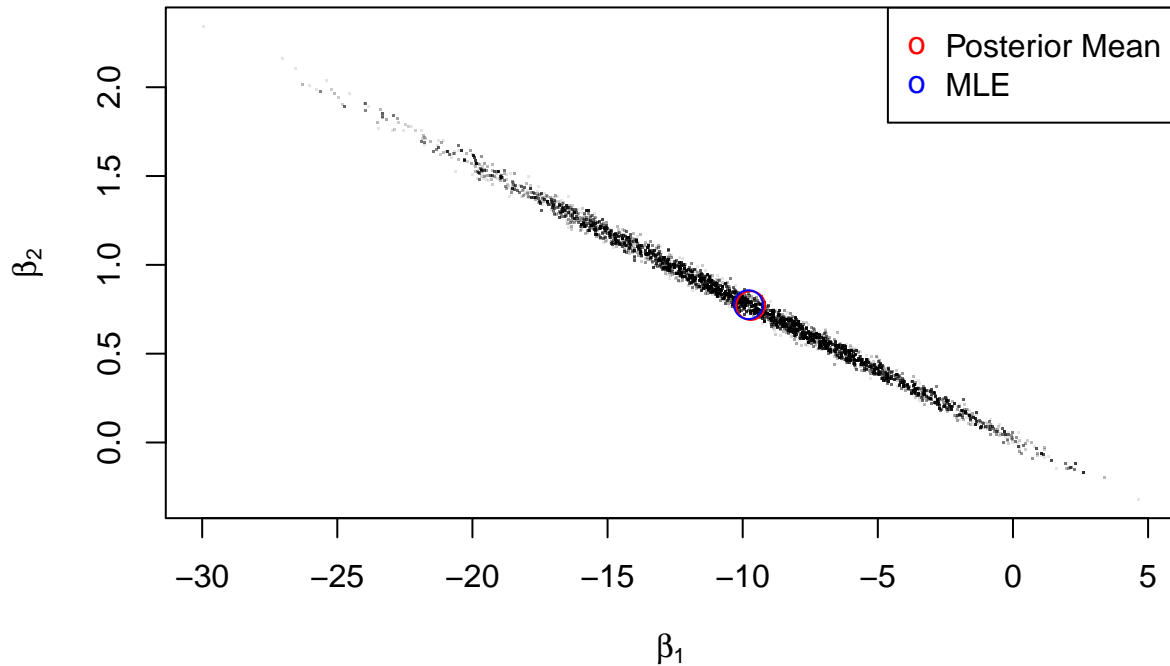
hist(BETA[, 2], breaks = 50, probability = TRUE,
     xlab = expression(beta[2]),
     main = expression(paste("Posterior distribution of ", beta[2])))
abline(v = post_mean_beta2, col = "black", lwd = 1, lty = 1)
abline(v = mle_beta2, col = "red", lwd = 1, lty = 1)
abline(v = ci_beta2, col = "blue", lwd = 2, lty = 3)
```



To aid with comparison, I will also plot the posterior samples of both parameters against each other and highlight where their posterior means and MLE occur.

```
plot(BETA[, 1], BETA[, 2],
     pch = ".", col = rgb(0, 0, 0, 0.1),
     xlab = expression(beta[1]), ylab = expression(beta[2]),
     main = "Posterior samples with MLE")
points(post_mean_beta1, post_mean_beta2, col = "red", cex=2)
points(mle_beta1, mle_beta2, col = "blue", cex=2)
legend("topright",
     legend = c("Posterior Mean", "MLE"),
     pch = c('o', 'o'), col = c("red", "blue"))
```


Posterior samples with MLE



Upon comparison, for each parameter, the posterior mean and the MLE are almost the same. At a macro-level, they are indistinguishable on the plots. Hence, one can conclude that both Frequentist and Bayesian methods provide similar results in this scenario.

(d) 30 points

Now the interest is on the nesting success probability for sparrow with a given wingspan x : $f_{\beta}(x) = \frac{e^{\beta_1 + \beta_2 x}}{1 + e^{\beta_1 + \beta_2 x}}$. Based on the samples from (1c), figure out how to compute posterior mean and 95% credible interval for $f_{\beta}(x)$ with $x \in [10, 16]$. Make a single plot including all these posterior quantities.

Answer

I will first generate values for $f_{\beta}(x) = \frac{e^{\beta_1 + \beta_2 x}}{1 + e^{\beta_1 + \beta_2 x}}$ for $x \in [10, 16]$. I am using 100 values in the grid for smoothness in the plot. I will then compute the posterior means from the resulting matrix, and generate the 95% credible interval for the f values by computing the appropriate quantile from the columns of the F_BETA matrix. I will then plot the posterior means and 95% credible intervals.

```
x <- seq(10,16,length.out = 100)
n_grid <- length(x)
n_samples <- nrow(BETA)

F_BETA <- matrix(0,nrow=n_samples,ncol=n_grid)
CI_F <- matrix(0,nrow=n_samples,ncol=2)
for (i in 1:n_samples) {
  F_BETA[i, ] <- exp(BETA[i, 1] + BETA[i, 2] * x) /
    (1 + exp(BETA[i, 1] + BETA[i, 2] * x))
}
```

```

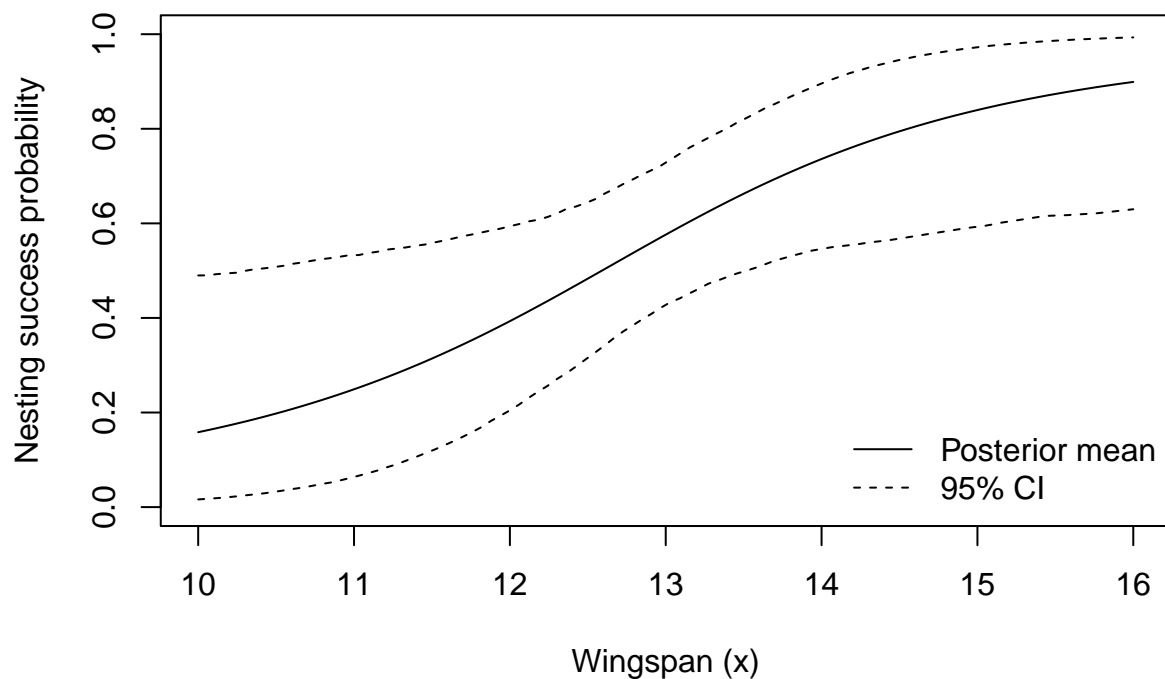
post_mean_f <- colMeans(F_BETA)

for (i in 1:length(x)) {
  CI_F[i, ] <- quantile(F_BETA[, i], c(0.025, 0.975))
}

plot(x, post_mean_f, type='l',
     xlab = "Wingspan (x)",
     ylab = "Nesting success probability",
     main="Posterior mean and 95% CI for sparrow nesting success probability",
     ylim = c(0,1))
lines(x, CI_F[1:100,1], lty=2)
lines(x, CI_F[1:100,2], lty=2)
legend("bottomright",
      legend = c("Posterior mean", "95% CI"),
      lty     = c(1, 2),
      bty     = "n")

```

Posterior mean and 95% CI for sparrow nesting success probability



We can observe an S-shaped logistic-like plot for the posterior means with the nesting success probability ranging from 0 to 1 as expected. Hence, my analysis of the trends in (1a) were accurate.