

# MATH 3190 Homework 6

Focus: Notes 8

Due March 30, 2024

Your homework should be completed in R Markdown or Quarto and Knitted to an html or pdf document. You will “turn in” this homework by uploading to your GitHub Math\_3190\_Assignment repository in the Homework directory.

Some of the parts in problems 1 and 2 require writing down some math-heavy expressions. You may either type it up using LaTeX style formatting in R Markdown, or you can write it by hand (neatly) and include pictures or scans of your work in your R Markdown document.

## Problem 1 (10 points)

Three airlines serve a small town in Ohio. Airline A has 52% of all scheduled flights, airline B has 35% and airline C has the remaining 13%. Their on-time rates are 85%, 67%, and 41%, respectively. A flight just left on-time. What is the probability that it was a flight of airline A?

## Problem 2 (13 points)

Suppose we have a data set with each observation  $x_i$  independent and identically exponentially distributed for  $i = 1, 2, \dots, n$ . That is,  $x_i \sim \text{Exp}(\lambda)$  where  $\lambda$  is the rate parameter. We would like to find a posterior (or at least a function proportional to it) for  $\lambda$ .

### Part a (5 points)

Write down the likelihood function (or a function proportional to it) in this situation. We would call this  $p(x|\lambda)$ .

### Part b (5 points)

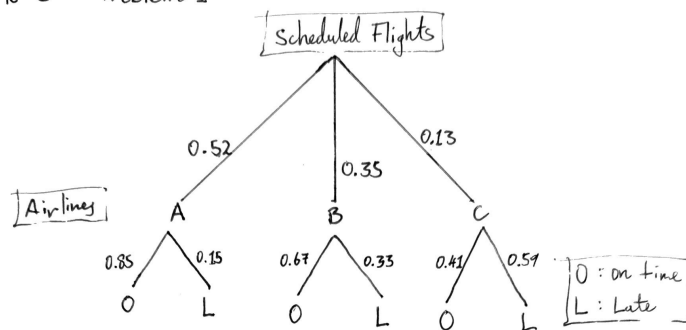
Now let  $\lambda$  have a normal prior with mean 0.1 and variance 1:  $\lambda \sim N(1/10, 1)$ . Use this and the likelihood from part a to write down a function that is proportional to the posterior of  $\lambda$  given  $\mathbf{x}$ . We call this  $p(\lambda|\mathbf{x})$ .

### Part c (3 points)

Which would be more appropriate here to obtain samples of  $\lambda$ , the Gibbs or Metropolis algorithm? Explain why. You may want to look on page 8 of Notes 8 in the conjugate prior table.

**Response:** The posterior and its kernel, in this case the entire expression, does not take the form of any properly defined conjugate. In this particular case, the Metropolis algorithm will be more appropriate, setting a symmetric distribution to sample from and using a random tolerance.

Homework 6 - Problem 1



$$\begin{aligned}
 P(A|O) &= \frac{P(O|A)P(A)}{P(O|A) \cdot P(A) + P(O|A^c) \cdot P(A^c)} \\
 &= \frac{(0.85)(0.52)}{(0.85)(0.52) + (0.67)(0.35) + (0.41)(0.13)} \\
 &= \frac{0.442}{0.442 + 0.2345 + 0.0533} \\
 &= \frac{0.442}{0.7298} = 0.60565
 \end{aligned}$$

⇒ 60.565% probability that an on-time flight was operated by airline A

Figure 1: Problem 1 response

Problem 2

$$\boxed{x_i \sim e^\lambda} \text{ for } \lambda \text{ rate}$$

$$\begin{aligned} \text{Part a. } p(x_i | \lambda) &= \prod_{i=1}^n \lambda e^{-\lambda x_i}, \quad x_i > 0 \\ &= (\lambda)^n \cdot e^{-\lambda \sum_{i=1}^n x_i} \\ &= (\lambda)^n \cdot e^{-\lambda \sum_{i=1}^n x_i} \quad \text{likelihood} \end{aligned}$$

$$\text{Part. b. } \lambda \sim N\left(\frac{1}{10}, 1\right) \stackrel{\text{Prior}}{=} \frac{1}{\sqrt{2\pi \cdot 1^2}} \cdot e^{\left\{-\frac{(\lambda - 0.1)^2}{2 \cdot 1^2}\right\}} = \frac{e^{\left\{-\frac{(\lambda - 0.1)^2}{2}\right\}}}{\sqrt{2\pi}}$$

$$\begin{aligned} \text{Posterior } p(\lambda | 0.1, 1) &\propto p(\lambda) p(x_i | \lambda) \\ &\propto \frac{1}{\sqrt{2\pi}} \cdot e^{\left\{-\frac{(\lambda - 0.1)^2}{2}\right\}} \cdot (\lambda)^n \cdot e^{\left\{-\lambda \cdot \sum_{i=1}^n x_i\right\}} \\ &\propto (\lambda)^n \cdot e^{\left\{-\frac{(\lambda - 0.1)^2}{2} - \lambda \cdot \sum_{i=1}^n x_i\right\}} \end{aligned}$$

*Gamma*

$$p(x | \alpha, \beta) = \frac{\beta^\alpha}{(\alpha-1)!} \cdot x^{\alpha-1} \cdot e^{-\beta x}$$

*Exponential*

$$p(x | \lambda) = \lambda \cdot e^{-\lambda x}$$

Figure 2: Problem 2 response

### Problem 3 (26 points)

Suppose we have the vector  $\mathbf{x} = c(1.83, 1.72, 2.13, 2.49, 0.90, 2.01, 1.51, 3.12, 1.29, 1.54, 2.94, 3.02, 0.93, 2.78)$  that we believe comes from a gamma distribution with shape of 10 and some rate  $\beta$ :  $x_i \sim \text{Gam}(10, \beta)$ . We will use sampling to obtain some information about  $\beta$ . Let's put a gamma prior on  $\beta$  with a shape of  $\alpha_0$  and a rate of 1:  $\beta \sim \text{Gam}(\alpha_0, 1)$ .

#### Part a (5 points)

Use the fact that this is a conjugate prior to write down what kind of distribution the posterior of  $\beta$ , which is  $p(\beta|\mathbf{x})$ , is.

Problem 3

Prior  $\beta \sim \Gamma(\alpha_0, 1)$

$\Leftrightarrow$  Posterior  $\beta|\mathbf{x} \sim \Gamma(\alpha_0 + 14(10), 1 + \sum_{i=1}^{14} x_i)$

$\Gamma(\alpha_0 + 140, 1 + 28.21)$

$\Gamma(\alpha_0 + 140, 29.21)$

$$p(\beta|\mathbf{x}) = \frac{(29.21)^{\alpha_0 + 140}}{(\alpha_0 + 140 - 1)!} \beta^{\alpha_0 + 140 - 1} \cdot e^{-29.21\beta}, \beta > 0$$

Figure 3: Problem 3 response

#### Part b (5 points)

Let  $\alpha_0 = 1$ . In an **R** code chunk, sample 10,000  $\beta$  values from the distribution you wrote down in part a using the `rgamma()` function and report the 95% credible interval for  $\beta$  using the 2.5th and 97.5th percentiles.

```
set.seed(2024)
x <- c(1.83, 1.72, 2.13, 2.49, 0.90, 2.01, 1.51,
       3.12, 1.29, 1.54, 2.94, 3.02, 0.93, 2.78)

alpha_0 <- 1

shape <- alpha_0 + length(x)*10
rate <- 1 + sum(x)

beta <- rgamma(10000, shape, rate)

quantile(beta, c(0.025, 0.975))

##      2.5%      97.5%
## 4.056263 5.660807
```

**Response:** There is a 95% chance that  $\beta$  lies in the interval  $[4.058, 5.627]$ .

### Part c (3 points)

Repeat part b with  $\alpha_0 = 10$ .

```
alpha_0 <- 10

shape <- alpha_0 + length(x)*10
rate <- 1 + sum(x)

beta <- rgamma(10000, shape, rate)

quantile(beta, c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 4.365546 5.985813
```

**Response:** There is a 95% chance that  $\beta$  lies in the interval  $[4.354, 5.992]$ .

### Part d (3 points)

Repeat part b with  $\alpha_0 = 100$ .

```
alpha_0 <- 100

shape <- alpha_0 + length(x)*10
rate <- 1 + sum(x)

beta <- rgamma(10000, shape, rate)

quantile(beta, c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 7.220025 9.300305
```

**Response:** There is a 95% chance that  $\beta$  lies in the interval  $[7.226, 9.300]$ .

### Part e (7 points)

Now suppose we have twice as much data (given in the **R** code chunk below). Repeat parts b, c, and d using this **x** vector instead and report the three 95% credible intervals. Note, this new vector **x** will change the shape and rate parameters used in the **rgamma()** functions.

```
x <- c(1.83, 1.72, 2.13, 2.49, 0.90, 2.01, 1.51, 3.12, 1.29, 1.54,
      2.94, 3.02, 0.93, 2.78, 2.76, 1.70, 1.42, 2.16, 1.07, 2.21,
      2.38, 2.27, 1.72, 1.44, 1.54, 1.72, 1.87, 1.39)

alpha_values <- c(1, 10, 100)

#credible intervals
credible_interval <- function(alpha_0, x) {
  shape <- alpha_0 + length(x) * 10
```

```

rate <- 1 + sum(x)
beta <- rgamma(10000, shape, rate)
interval <- quantile(beta, c(0.025, 0.975))
return(interval)
}

#intervals for each alpha_0
results <- map_df(alpha_values, ~{
  alpha_0 <- .x
  interval <- credible_interval(alpha_0, x)
  data.frame(alpha_0 = alpha_0,
             interval_lower = interval[1], # Extract lower bound
             interval_upper = interval[2]) # Extract upper bound
})

#intervals as strings
results$interval_str <- sprintf("%.3f, %.3f", results$interval_lower, results$interval_upper)

results |>
  rownames_to_column(var = "row_id") |>
  select(-c(interval_lower, interval_upper, row_id)) |> #retirer la merde
  kable(col.names = c("a_0", "Credible Interval")) |>
  kable_styling(full_width = FALSE) |>
  column_spec(2, bold = TRUE)

```

a_0	Credible Interval
1	[4.535, 5.740]
10	[4.690, 5.928]
100	[6.234, 7.650]

### Part f (3 points)

In this problem, the true  $\beta$  value is 5. Write a sentence or two about the effect adding more data has to these credible intervals by comparing the intervals from parts b-d to the intervals from part e.

**Response:** The increased sample data reduced the range of each credible interval, and in the case of the  $\alpha_0 = 1$  and 10 the interval was closer to and more centered over the true  $\beta = 5$ . Where  $\alpha_0 = 100$  the interval [7.226, 9.300] was not re-centered over 5, but reduced and drawn more closely to the true  $\beta$  by the interval [6.240, 7.634].

## Problem 4 (51 points)

Let's apply the Bayesian framework to a regression problem. In the GitHub data folder, there is a file called `treeseeds.txt` that contains information about species of tree, the count of seeds it produces, and the average weight of those seeds in *mg*.

### Part a (3 points)

Read in the `treeseeds.txt` file and take the log of the counts and weights. Fit an OLS regression model using  $\log(\text{weight})$  to predict  $\log(\text{count})$ .

```
treeseeds <- read_csv(
  paste0("/Users/bram/Documents/Math_3190/Homework_6_Data/treeseeds.txt"))

## Rows: 19 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (1): species
## dbl (2): count, weight
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

treeseeds_log <- treeseeds |>
  mutate(ln_count = log(count), ln_weight = log(weight)) |>
  select(-count, -weight)

treeseeds_mod <- glm(ln_count ~ ln_weight, data = treeseeds_log, family = "gaussian")

summary(treeseeds_mod)

##
## Call:
## glm(formula = ln_count ~ ln_weight, family = "gaussian", data = treeseeds_log)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.37903    0.39720  23.613 1.95e-14 ***
## ln_weight   -0.51491    0.07185  -7.166 1.58e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.8766687)
##
##      Null deviance: 59.921  on 18  degrees of freedom
## Residual deviance: 14.903  on 17  degrees of freedom
## AIC: 55.305
##
## Number of Fisher Scoring iterations: 2
```

### Part b (15 points)

We will walk through the mathematics of obtaining the posterior together here since this problem will focus on coding the Metropolis algorithm. Assuming the true errors are normal with mean 0 and variance  $\sigma^2$ ,  $\epsilon_i \sim N(0, \sigma^2)$ , it can be shown that each  $y_i$  has the distribution

$$p(y_i | x_i, \beta_0, \beta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right)$$

So, we can write the likelihood is

$$p(y_i|x_i, \beta_0, \beta_1, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right)$$

where  $y_i$  is the log(count) for observation  $i$  and  $x_i$  is the log(weight) for observation  $i$ . Note that here we think of  $\mathbf{y}$  as being random and  $\mathbf{x}$  as being fixed. We could, in theory, think of the vector  $\mathbf{x}$  as also being random and put a prior on it. But we won't do that here.

Now, let's just put uniform priors on  $\beta_0$  and  $\beta_1$  so the priors are proportional to 1. Also, let's assume  $\sigma^2 = 1$ . This seems reasonable since  $s_e^2$ , the MSE, is 0.877. Of course, we could put a prior on  $\sigma^2$  as well and sample it too, but we will focus on only sampling  $\beta_0$  and  $\beta_1$ .

Now, with those uniform priors, and plugging in 1 for  $\sigma^2$ , we have that the joint posterior of  $\beta_0$  and  $\beta_1$  is:

$$p(\beta_0, \beta_1|\mathbf{x}, \mathbf{y}) \propto \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2\right) = f(\beta_0, \beta_1|\mathbf{x}, \mathbf{y}).$$

Then, we can take the log to get

$$\ln(f(\beta_0, \beta_1|\mathbf{x}, \mathbf{y})) = -\frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2.$$

Our goal now is to obtain samples of  $\beta_0$  and  $\beta_1$ . Let's use the Metropolis algorithm to do this. Using the log of the function proportional to the joint posterior of  $\beta_0$  and  $\beta_1$ ,  $\ln(f(\beta_0, \beta_1|\mathbf{x}, \mathbf{y}))$ , write a Metropolis algorithm in **R**. For  $\beta_0$ , you can use a normal proposal distribution centered at the previous value,  $\beta_0^{(i)}$ , with a standard deviation of 0.8 and for  $\beta_1$ , you can use a normal proposal distribution centered at the previous value,  $\beta_1^{(i)}$ , with a standard deviation of 0.1. The starting values don't matter too much, but we can use  $\beta_0^{(0)} = 10$  and  $\beta_1^{(0)} = -0.5$ . It may be useful to look at the **Notes 8 Script.R** file that is on GitHub in the Notes 8 folder and is on Canvas.

Obtain at least 10,000 samples (set a seed, please) and plot the chains for  $\beta_0$  and  $\beta_1$ . For this problem, include:

1. The plot for the  $\beta_0$  chain.
2. The plot for the  $\beta_1$  chain.
3. The 95% credible interval for  $\beta_0$  based on the 2.5th and 97.5th percentiles.
4. The 95% credible interval for  $\beta_1$  based on the 2.5th and 97.5th percentiles.

```
joint_function <- function(y, x, b_0, b_1) {  
  arg <- y - b_0 - b_1 * x  
  arg2 <- arg^2  
  result <- (-1/2) * sum(arg2)  
  return(result)  
}  
  
set.seed(2024)  
  
n_samps <- 10000  
beta_0 <- rep(0, n_samps)  
beta_1 <- rep(0, n_samps)  
  
beta_0[1] <- 10  
beta_1[1] <- -0.5
```



```

n <- length(treeseeds)

for (i in 1:(n_samps - 1)) {
  beta_0_star <- rnorm(1, beta_0[i], 0.8)
  beta_1_star <- rnorm(1, beta_1[i], 0.1)

  # using the natural log of the proportional joint posterior distribution of  $\beta_0, \beta_1$  as function:

  ##for the ratio R numerator  $f(\emptyset^*/x)$  (is logged for difference)
  ln_f1 <- joint_function(treeseeds_log$ln_count, treeseeds_log$ln_weight, beta_0_star, beta_1_star)

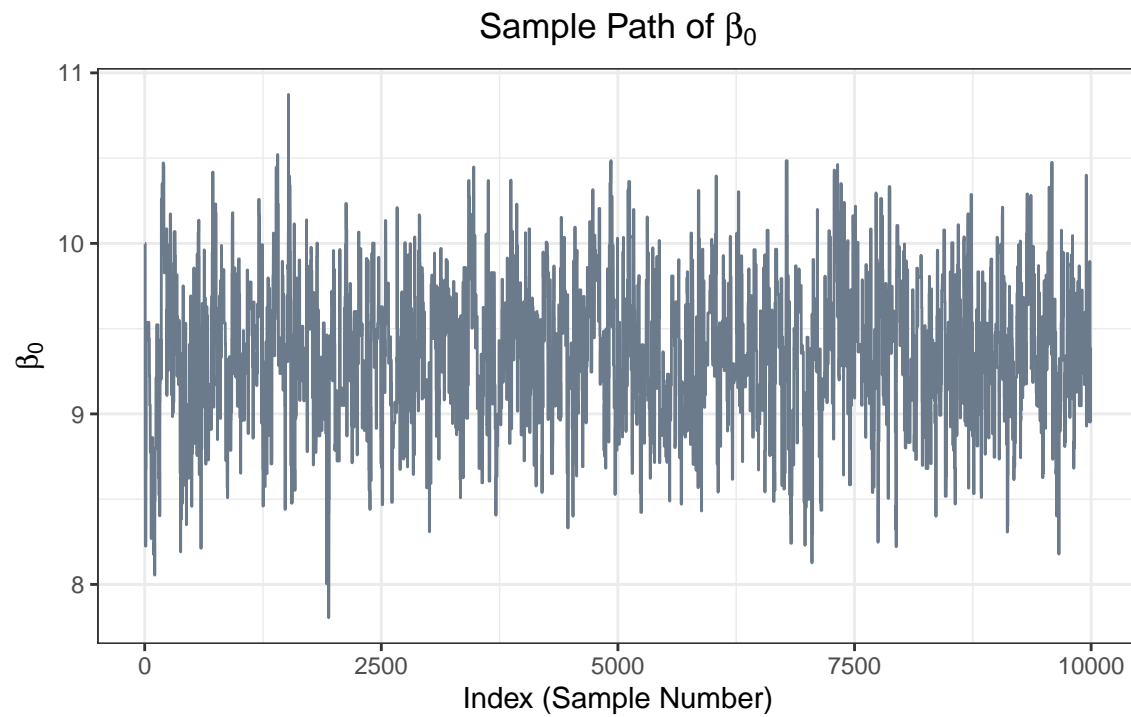
  ## for the ratio R denominator  $f(\emptyset^i/x)$  (is logged for difference)
  ln_f2 <- joint_function(treeseeds_log$ln_count, treeseeds_log$ln_weight, beta_0[i], beta_1[i])

  if (log(runif(1)) < (ln_f1 - ln_f2)) {
    beta_0[i+1] <- beta_0_star
    beta_1[i+1] <- beta_1_star
  } else {
    beta_0[i+1] <- beta_0[i]
    beta_1[i+1] <- beta_1[i]
  }
}

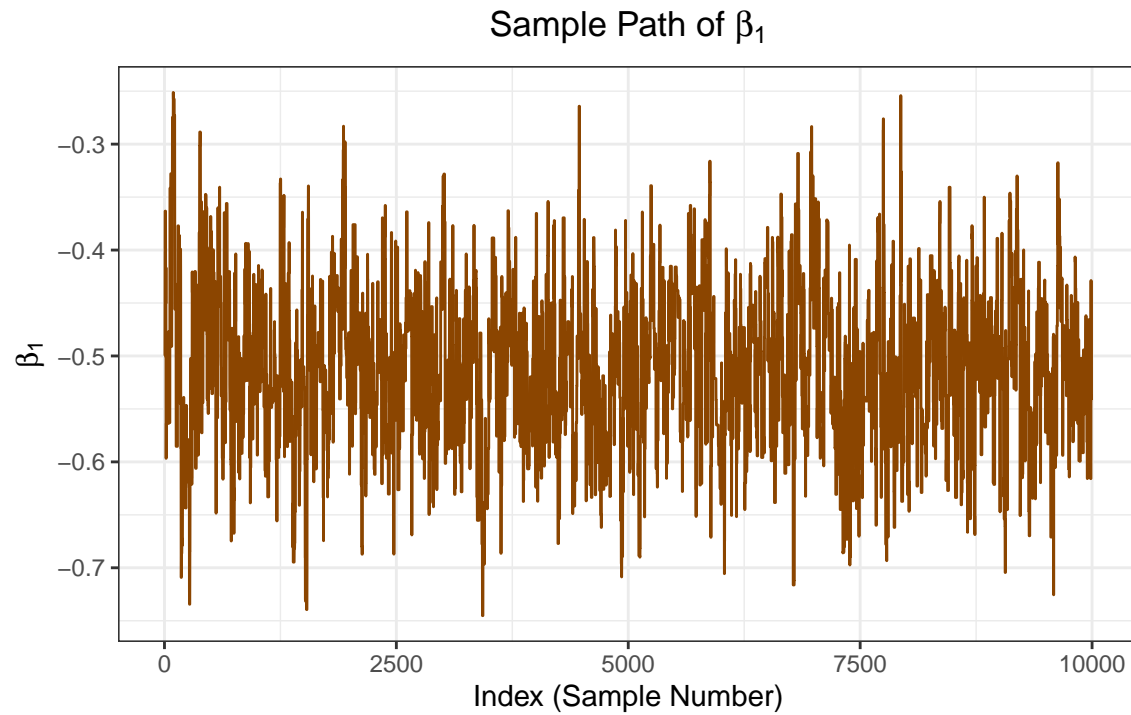
}

# Sample path plots
ggplot(data.frame(x = 1:n_samps, beta_0)) +
  geom_line(aes(x = x, y = beta_0), color = "slategray4", linetype = "solid") +
  labs(title = expression(paste("Sample Path of ", beta[0])),
       x = "Index (Sample Number)", y = expression(beta[0]),
       caption = "", nudge_y = 0.1) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

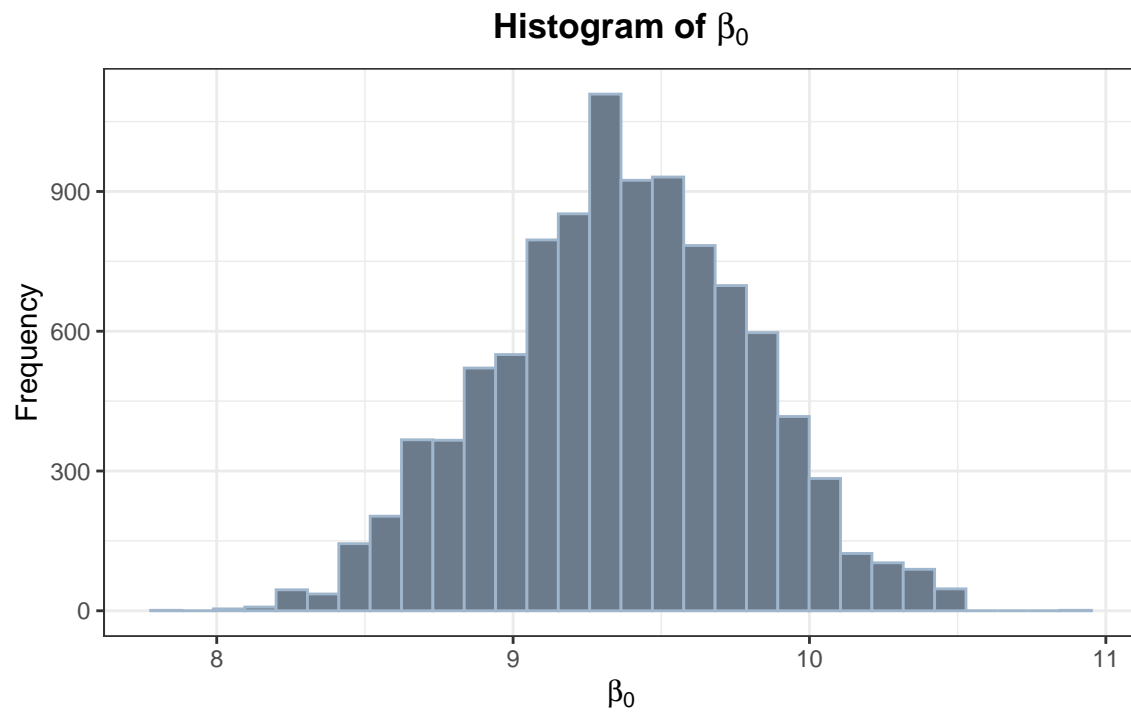
```



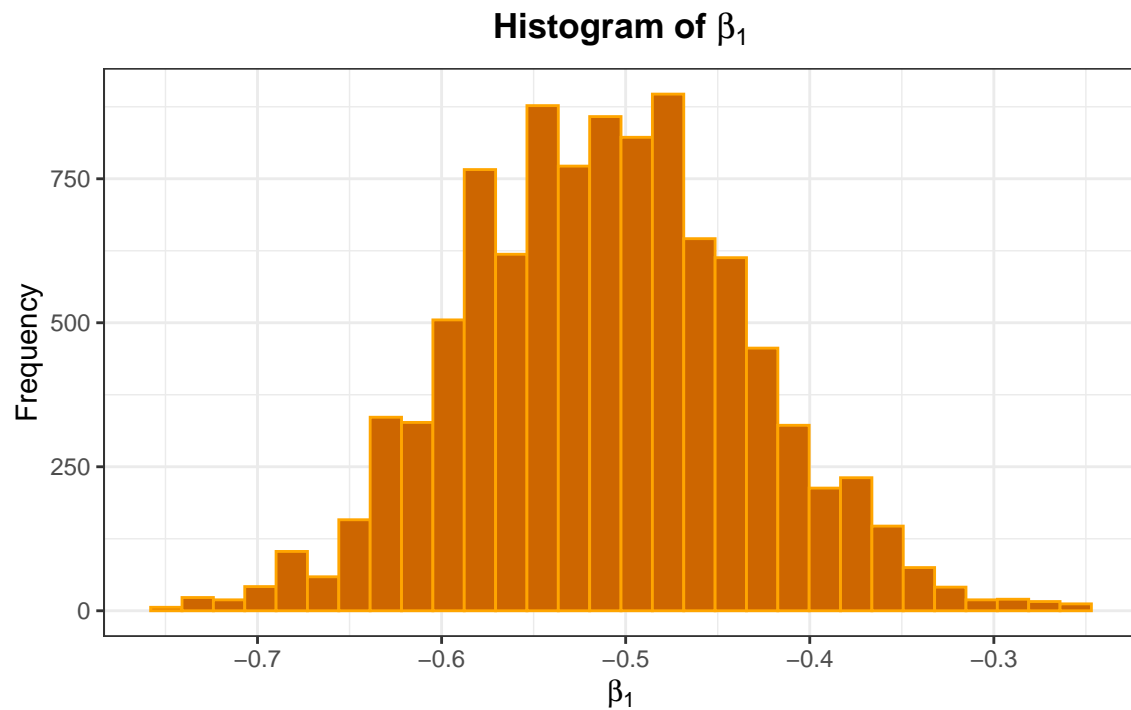
```
ggplot(data.frame(x = 1:n_samps, beta_1)) +
  geom_line(aes(x = x, y = beta_1), color = "darkorange4", linetype = "solid") +
  labs(title = expression(paste("Sample Path of ", beta[1])),
       x = "Index (Sample Number)", y = expression(beta[1]),
       caption = "", nudge_y = 0.1) +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Histogram plots
ggplot(data.frame(beta_0)) +
  geom_histogram(aes(x = beta_0), bins = 30, color = "slategray3", fill = "slategray4") +
  labs(title = expression(paste(bold("Histogram of "), beta[0])), x = expression(beta[0]), y = "Frequency") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(data.frame(beta_1)) +
  geom_histogram(aes(x = beta_1), bins = 30, color = "orange1", fill = "darkorange3") +
  labs(title = expression(paste(bold("Histogram of "), beta[1])),
       x = expression(beta[1]), y = "Frequency",
       caption = "") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
#quantiles for betas
quantile(beta_0, c(0.025, 0.975))
```

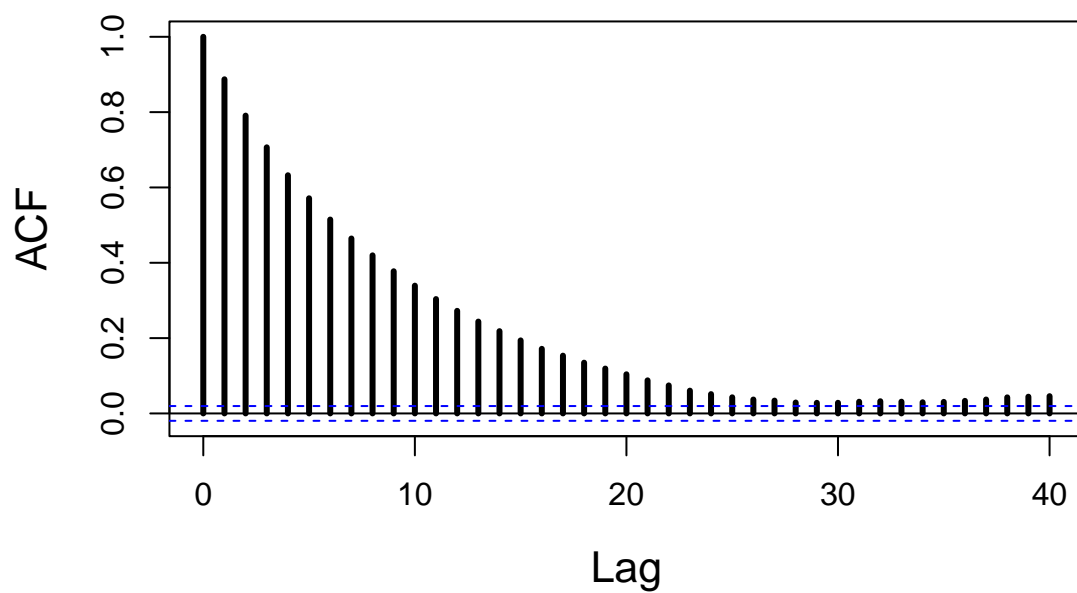
```
##      2.5%      97.5%
##  8.528203 10.199103
```

```
quantile(beta_1, c(0.025, 0.975))
```

```
##      2.5%      97.5%
## -0.6598421 -0.3586130
```

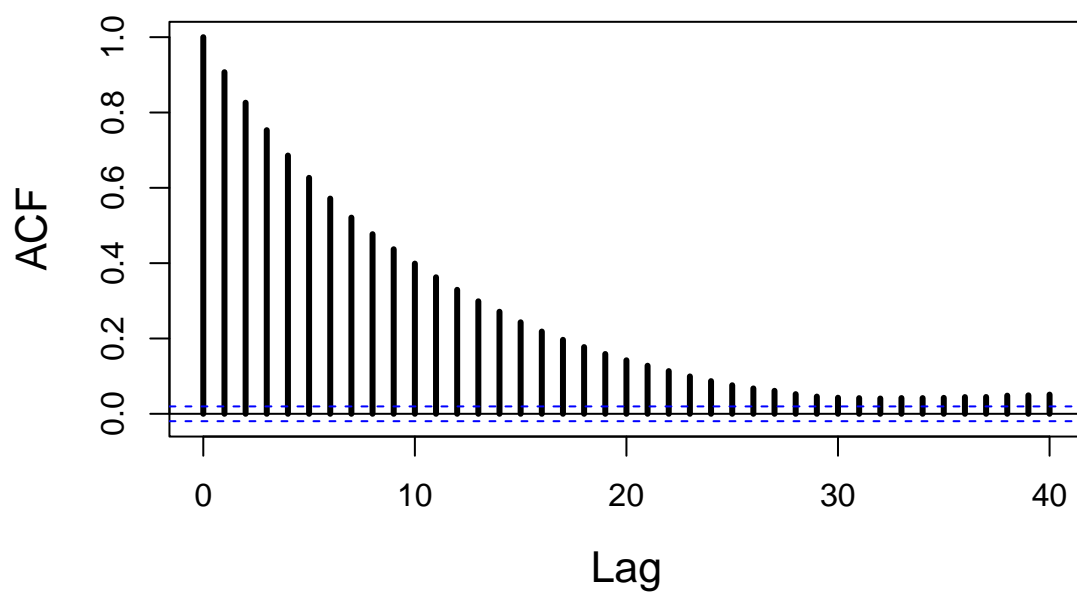
```
#autocorrelation plots for betas
acf(beta_0, main = expression(paste("Autocorrelation of ", beta[0], " Chain")), mgp = c(2.7, 1, 0), cex = 0.8)
```

Autocorrelation of  $\beta_0$  Chain



```
acf(beta_1, main = expression(paste("Autocorrelation of ", beta[1], " Chain")), mgp = c(2.7, 1, 0), cex
```

Autocorrelation of  $\beta_1$  Chain



### Part c (3 points)

Based on the plots of the chains from part b, does it look like the Metropolis sampling worked fairly well?

**Response:** The graph of the sampling chain does not exhibit strong directional patterns over the 10,000 iterations, suggesting that there are few iterations which depend on many preceding it. We infer that the Metropolis sampling worked relatively well.

### Part d (4 points)

Interpret both of the credible intervals from part b.

```
cred_intervals <- data.frame(  
  Parameter_Variable = c('β_0', 'β_1'),  
  Lower_2.5 = c(quantile(beta_0, 0.025), quantile(beta_1, 0.025)),  
  Upper_97.5 = c(quantile(beta_0, 0.975), quantile(beta_1, 0.975))  
)  
  
kable(cred_intervals, format = "html", align = "c") |>  
  kable_styling(full_width = FALSE)
```

Parameter\_Variable

Lower\_2.5

Upper\_97.5

β\_0

8.5282034

10.199103

β\_1

-0.6598421

-0.358613

**Response:** There is a 95% chance that the true natural log of seed counts (intercept  $\beta_0$ ) lies in the interval [8.528, 10.199] when the log of seed weights is zero.

There is a 95% chance that the true change in the natural log of seed counts (slope  $\beta_1$ ) lies in the interval [-0.660, -0.359] for each unit change in the natural log of seed weights.

### Part e (5 points)

Find and report the integrated autocorrelation time for the  $\beta_0$  and  $\beta_1$  chains. Each chain will have their own  $\hat{\tau}_{int}$  value, so you should report two (although they will be similar).

```
#ESS calculations for betas  
ess_beta0 <- 1 + 2 * sum( abs(acf(beta_0, lag.max = 100, plot = F)$acf) )  
K_ESS_1 <- n_samps/ess_beta0  
  
ess_beta1 <- 1 + 2 * sum( abs(acf(beta_1, lag.max = 100, plot = F)$acf) )  
print(ess_beta1)
```

```
## [1] 25.78817
```

```
K_ESS_2 <- n_samps/ess_beta1

print(paste0('Integrated Autocorrelation Time for  $\beta_0$  is ',round(ess_beta0, 3)))
```

```
## [1] "Integrated Autocorrelation Time for  $\beta_0$  is 22.709"
```

```
print(paste0('Integrated Autocorrelation Time for  $\beta_1$  is ',round(ess_beta1, 3)))
```

```
## [1] "Integrated Autocorrelation Time for  $\beta_1$  is 25.788"
```

### Part f (3 points)

Based on the integrated autocorrelation time for the  $\beta_0$  and  $\beta_1$  chains, how many MCMC samples would you need to generate to get the equivalent of 10,000 independent samples?

```
equiv_ss1 <- n_samps * ess_beta0
equiv_ss2 <- n_samps * ess_beta1

print(paste(" $\beta_0$ :", round(equiv_ss1,1), " $\beta_1$ :", round(equiv_ss2,1)))
```

```
## [1] " $\beta_0$ : 227089.8  $\beta_1$ : 257881.7"
```

Considering the  $\hat{\tau}_{int}$  for  $\beta_1$ , to obtain the equivalent of 10,000 samples would require the number of MCMC samples to be at least 257,882.

### Part g (3 points)

Let's compare these credible intervals to some other intervals. First, obtain the 95%  $t$  confidence intervals for  $\beta_0$  and  $\beta_1$  just using the `confint()` function and report them here.

```
ci_seed_model <- confint(treeseeds_mod)

## Waiting for profiling to be done...

row_names <- c(" $\beta_0$  (Intercept)", " $\beta_1$  (ln_weight)")
rownames(ci_seed_model) <- row_names

colnames(ci_seed_model) <- c("2.5 %", "97.5 %")

kable(ci_seed_model, format = "html") |>
  kable_styling(full_width = FALSE) |>
  add_header_above(c("Parameter Confidence Intervals" = 3))
```

Parameter Confidence Intervals

2.5 %

97.5 %

$\beta_0$  (Intercept)



```
8.6005353
10.1575170
β_1 (ln_weight)
-0.6557376
-0.3740727
```

### Part h (10 points)

Now let's obtain confidence intervals using bootstrapping in a similar way we did with regularization in Notes 7 and HW 4 (this is known as bootstrapping the cases). Set a seed and then using at least 10,000 bootstrap samples, report the 95% percentile confidence intervals for  $\beta_0$  and  $\beta_1$  using the `quantile()` function on the values of  $\beta_0$  and  $\beta_1$  that you obtained in the bootstrap.

```
set.seed(2024)

n_boots <- 10000

beta_matrix <- matrix(rep(0, 2 * n_boots), nrow=n_boots)

for(i in 1:n_boots){

  index <- sample(1:nrow(treeseeds_log), nrow(treeseeds_log), replace=T)

  beta_matrix[i,] <- coef(
    glm(ln_count~ln_weight, family=gaussian, data=treeseeds_log[index, ])
  )
}

quantile(beta_matrix[,1], c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 8.422312 10.281302
```

```
quantile(beta_matrix[,2], c(0.025, 0.975))
```

```
##      2.5%      97.5%
## -0.6628756 -0.3540499
```

**Response:** We can say with 95% confidence that the true natural log of seed counts (intercept  $\beta_0$ ) lies in the interval [8.601, 10.158] when the log of seed weights is zero.

We can say with 95% that the true change in the natural log of seed counts (slope  $\beta_1$ ) lies in the interval [-0.656, -0.374] for each unit change in the natural log of seed weights.

### Part i (5 points)

Write a couple sentences comparing all of the intervals in parts b, g, and h.

**Response:** The count of tree seeds as a random variable was identified as being normally distributed, that is  $y|x, \beta_0, \beta_1, \sigma$  approximately follows a Gaussian probability distribution. The intervals in each case, for the

Bayesian inference performed with an MCMC chain as for the confidence intervals obtained from a Student's  $t$ -distribution, the credibility intervals and the confidence intervals were very similar. Were the random variable not normally distributed, there would be some skew in particular between the Bayesian interval and the Student interval, even if some similarity with the bootstrapped interval subsisted.