Take Home Final

Gabe Mancino

4/25/2018

This is a take-home exam, and is to be completed on your own. Any evidence of collaboration will result in severe penalization for all collaborators. Submit your responses in a Word or pdf document compiled by R Markdown, along with your .Rmd source, to D2L by 11:59 pm on May 3, 2018.

1. (30 points) (Modified from Dr. Bergen's MS qualifying exam, June 2010.) Consider the simple linear regression problem:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, ..., n$$

The x_i are fixed and known and are mean-centered, implying that $\sum_{i=1}^n x_i = 0$. The error terms $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ with known σ^2. (Note that all of this is equivalent to saying $Y_i \overset{iid}{\sim} N(\beta_0 + \beta_1 x_i, \sigma^2)$). The regression parameters β_0 and β_1 are unknown, and the target of inference is $\theta = \beta_1^2$.

A. (5 points) Find $\hat{\theta}_{MLE}$.

Solution.

Since MLE's are invariant, finding $\hat{\beta}_{1,MLE}$ and squaring this will give $\hat{\theta}_{MLE}$, so begin by simplifying $L(\beta_1)$:

$$L(\beta_1) = \prod_{i=1}^n \left(\frac{1}{2\pi\sigma^2}\right)^{1/2} e^{\frac{1}{2\sigma^2}(y_i - (\beta_0 + \beta_1 x_i))^2} = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} e^{\frac{1}{2\sigma^2}\sum(y_i - (\beta_0 + \beta_1 x_i))^2}.$$

Focus on the exponent by foiling:

$$\frac{1}{2\sigma^2}\sum(y_i - (\beta_0 + \beta_1 x_i))^2 = \frac{1}{2\sigma^2}\sum(\beta_0^2 + 2\beta_0\beta_1 x_i + \beta_1^2 x_i^2 - 2\beta_0 y_i - 2\beta_1 x_i y_i + y_i^2).$$

Now taking the natural log yields:

$$\begin{aligned}
\ln L(\beta_1) &= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum(\beta_0^2 + 2\beta_0\beta_1 x_i + \beta_1^2 x_i^2 - 2\beta_0 y_i - 2\beta_1 x_i y_i + y_i^2) \\
&= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(n\beta_0^2 + \beta_0\beta_1\sum x_i + \beta_1^2\sum x_i^2 - 2\beta_0\sum y_i - 2\beta_1\sum x_i y_i + \sum y_i^2).
\end{aligned}$$

Taking the derivative with respect to β_1 gives:

$$\frac{\partial}{\partial\beta_1}\ln L(\beta_1) = \frac{\sum x_i y_i}{\sigma^2} - \frac{\beta_1\sum x_i^2}{\sigma^2} \overset{set}{=} 0.$$

Thus $\hat{\beta}_1 = \frac{\sum y_i x_i}{\sum x_i^2}$ and since MLE's are invariant,

$$\hat{\theta}_{MLE} = \left(\frac{\sum y_i x_i}{\sum x_i^2}\right)^2.$$

□

B. (5 points) Find the bias of $\hat{\theta}_{MLE}$ for estimating θ.

Solution.

First, note that $E[\left(\frac{\sum y_i x_i}{\sum x_i^2}\right)^2] = \frac{1}{(\sum x_i^2)^2} E[(\sum y_i x_i)^2]$. Now, we make use of the fact that $E[X^2] = Var[X] - E[X]^2$. First, computing $E[\sum y_i x_i] = x_1(\beta_0 + \beta_1 x_1) + x_2(\beta_0 + \beta_1 x_2) + \cdots + x_n(\beta_0 + \beta_1 x_n) = \beta_0 \sum x_i + \beta_1 \sum x_i^2 = \beta_1 \sum x_i^2$. Squaring this yields $E[\sum x_i y_i]^2 = \beta_1^2 (\sum x_i^2)^2$. Second, computing $Var[\sum y_i x_i] = \sum x_i^2 Var(y_i)$ (notice the covariances are zero since y_i is independent from y_j for $i \neq j$), then, $Var[y_i] = \sigma^2$ for all y_i so $Var[\sum y_i x_i] = \sigma^2 \sum x_i^2$. Combining the above information:

$$\frac{1}{(\sum x_i^2)^2} E[(\sum y_i x_i)^2] = \frac{1}{(\sum x_i^2)^2}\left(\sigma^2 \sum x_i^2 + \beta_1^2 (\sum x_i^2)^2\right) = \frac{\sigma^2}{\sum x_i^2} + \beta_1^2.$$

Thus the bias of $\hat{\theta}_{MLE}$ is $\frac{\sigma^2}{\sum x_i^2}$.

□

C. (5 points) Derive a crude lower bound for the variance of $\hat{\theta}_{MLE}$ by treating β_0 as known.

Solution.

Using the definition of the Cramer-Rao Lower Bound and the fact that $Var(Y + b) = Var(Y)$ for all constants b, define $\tau(\theta) := \beta_1^2 + \frac{\sigma^2}{\sum x_i^2}$, then $\hat{\theta}_{MLE}$ is unbiased for $\hat{\tau}(\theta)$. Thus,

$$Var(\hat{\theta}_{MLE}) = Var(\hat{\tau}(\theta)) \geq \frac{\tau'(\theta)^2}{-nE[\frac{\partial^2}{\partial \beta_1^2} \ln L(y; \beta_1)]}$$

Computing $\tau'(\theta) = 2\beta_1$ and $I(\beta_1) = \frac{n \sum x_i^2}{\sigma^2}$ we find that a lower bound on the variance of $\hat{\theta}_{MLE}$ is $\frac{4\beta_1^2 \sigma^2}{n \sum x_i^2}$ where in practice β_1 would need to be estimated.

□

D. (3 points) Derive an unbiased estimator for θ by a simple modification to $\hat{\theta}_{MLE}$ (call it $\hat{\theta}_{UB}$).

Solution.

Simply subtract $\frac{\sigma^2}{\sum x_i^2}$ from $\hat{\theta}_{MLE}$ to receive

$$\hat{\theta}_{UB} = \hat{\theta}_{MLE} - \frac{\sigma^2}{\sum x_i^2}.$$

□

E. (3 points) Identify a shortcoming of $\hat{\theta}_{UB}$ and suggest an improvement. Let $\hat{\theta}_{IMP}$ notate your suggested improvement.

Solution.

Notice that if σ^2 is large (or $\sum x_i^2$ is sufficiently small) then $\hat{\theta}_{UB} < 0$. This is bad since the quantity θ is estimating is $\beta_1^2 \in [0, \infty)$. So defining

$$\hat{\theta}_{IMP} = \begin{cases} 0 & \text{if } \frac{\sigma^2}{\sum x_i^2} > \hat{\theta}_{MLE} \\ \hat{\theta}_{UB} & \text{otherwise} \end{cases}$$

will be an improved estimator.

□

F. (9 points) Let $\beta_0 = 0$ and $\sigma^2 = 1$. Given n, suppose there are $n/5$ x_i each at $\{-2, -1, 0, 1, 2\}$. Consider all 6 combinations of $n \in \{10, 20, 100\}$ and $\beta_1 \in \{0.5, 2\}$. Carry out a simulation study to compare the MSE of your three estimators $\hat{\theta}_{MLE}$, $\hat{\theta}_{UB}$, and $\hat{\theta}_{IMP}$. Summarize your simulation results in a table. Full credit will only be given if your results are rounded to a reasonable number of digits. Comment on which estimator is best.

Solution.

Consider the following code and tables.

```
# Function for computing Thetas for Beta_1 = 0.5
theta.hats.0.5 <- function(n) {
    xi <- c(-2, -1, 0, 1, 2)
    samp <- c()
    for (i in 1:5) {
        samp <- c(samp, rnorm(n/5, mean = 0.5 * xi[i], sd = 1))
    }
    xi <- rep(xi, each = n/5)
    num <- c()
    for (i in 1:n) {
        num[i] <- samp[i] * xi[i]
    }
    theta.mle <- (sum(num)/sum(xi^2))^2
    theta.ub <- theta.mle - (1/sum(xi^2))
    theta.imp <- ifelse((1/sum(xi^2)) > theta.mle, 0, theta.ub)
    return(c(Theta.MLE = theta.mle, Theta.UB = theta.ub, Theta.IMP = theta.imp))
}
# MSE function for Beta_1 = 0.5
mse.0.5 <- function(estimator) {
    bias <- mean(estimator) - 0.5^2
    var <- var(estimator)
    mse <- var + (bias)^2
    return(mse)
}
# Replication and Data Frames for Beta_1 = 0.5
r <- 10000
many.thetahats.0.5_n10 <- replicate(r, theta.hats.0.5(10))
many.thetahats.0.5_n20 <- replicate(r, theta.hats.0.5(20))
many.thetahats.0.5_n100 <- replicate(r, theta.hats.0.5(100))
df.mse.0.5.10 <- round(data.frame(MSE.Theta_MLE = mse.0.5(many.thetahats.0.5_n10[1,
    ]), MSE.Theta_UB = mse.0.5(many.thetahats.0.5_n10[2, ]),
    MSE.Theta.IMP = mse.0.5(many.thetahats.0.5_n10[3, ])), digits = 3)
df.mse.0.5.20 <- round(data.frame(MSE.Theta_MLE = mse.0.5(many.thetahats.0.5_n20[1,
    ]), MSE.Theta_UB = mse.0.5(many.thetahats.0.5_n20[2, ]),
    MSE.Theta.IMP = mse.0.5(many.thetahats.0.5_n20[3, ])), digits = 3)
df.mse.0.5.100 <- round(data.frame(MSE.Theta_MLE = mse.0.5(many.thetahats.0.5_n100[1,
    ]), MSE.Theta_UB = mse.0.5(many.thetahats.0.5_n100[2, ]),
    MSE.Theta.IMP = mse.0.5(many.thetahats.0.5_n100[3, ])), digits = 3)

# Function for computing Thetas for Beta_1 = 2
theta.hats.2 <- function(n) {
    xi <- c(-2, -1, 0, 1, 2)
    samp <- c()
    for (i in 1:5) {
        samp <- c(samp, rnorm(n/5, mean = 2 * xi[i], sd = 1))
```

```
    }
    xi <- rep(xi, each = n/5)
    num <- c()
    for (i in 1:n) {
        num[i] <- samp[i] * xi[i]
    }
    theta.mle <- (sum(num)/sum(xi^2))^2
    theta.ub <- theta.mle - (1/sum(xi^2))
    theta.imp <- ifelse((1/sum(xi^2)) > theta.mle, 0, theta.ub)
    return(c(Theta.MLE = theta.mle, Theta.UB = theta.ub, Theta.IMP = theta.imp))
}
# MSE function for Beta_1 = 2
mse.2 <- function(estimator) {
    bias <- mean(estimator) - 2^2
    var <- var(estimator)
    mse <- var + (bias)^2
    return(mse)
}
# Replication and Data Frames for Beta_1 = 2
many.thetahats.2_n10 <- replicate(r, theta.hats.2(10))
many.thetahats.2_n20 <- replicate(r, theta.hats.2(20))
many.thetahats.2_n100 <- replicate(r, theta.hats.2(100))
df.mse.2.10 <- round(data.frame(MSE.Theta_MLE = mse.2(many.thetahats.2_n10[1,
    ]), MSE.Theta_UB = mse.2(many.thetahats.2_n10[2, ]), MSE.Theta.IMP = mse.2(many.thetahats.2_n10[3,
    ])), digits = 3)
df.mse.2.20 <- round(data.frame(MSE.Theta_MLE = mse.2(many.thetahats.2_n20[1,
    ]), MSE.Theta_UB = mse.2(many.thetahats.2_n20[2, ]), MSE.Theta.IMP = mse.2(many.thetahats.2_n20[3,
    ])), digits = 3)
df.mse.2.100 <- round(data.frame(MSE.Theta_MLE = mse.2(many.thetahats.2_n100[1,
    ]), MSE.Theta_UB = mse.2(many.thetahats.2_n100[2, ]), MSE.Theta.IMP = mse.2(many.thetahats.2_n100[3
    ])), digits = 3)
```

$\beta_1 = 0.5$	$n = 10$	$n = 20$	$n = 100$
$MSE(\hat{\theta}_{MLE})$	0.054	0.028	0.005
$MSE(\hat{\theta}_{UB})$	0.052	0.027	0.005
$MSE(\hat{\theta}_{IMP})$	0.05	0.027	0.005

$\beta_1 = 2$	$n = 10$	$n = 20$	$n = 100$
$MSE(\hat{\theta}_{MLE})$	0.795	0.409	0.079
$MSE(\hat{\theta}_{UB})$	0.794	0.408	0.078
$MSE(\hat{\theta}_{IMP})$	0.794	0.408	0.078

By the above tables, we find that $\hat{\theta}_{IMP}$ is the best estimator with respect to minimizing MSE.

$\square$

2. (35 points) (Modified from Gelman et al *Bayesian Data Analysis* 3ed, p 45). Suppose that causes of death are reviewed in detail for a city of size 200,000 in the United States for a single year. Let Y represent the number of persons, out of a population of 200,000, that died of asthma within a year. A Poisson model is often used for epidemiological data of this form; specifically, $Y \sim POI(2\theta)$, where θ represents the true rate of deaths from asthma, per 100,000 population. Suppose that within the year, $Y = 3$.

A. (2 points) What is $\hat{\theta}_{MLE}$?

Solution.

To find $\hat{\theta}_{MLE}$, maximize $L(\theta) = \prod_{i=1}^{n} e^{-2\theta} \frac{2\theta^{y_i}}{y_i!} = e^{-2n\theta} \frac{(2\theta)^{\sum y_i}}{\prod y_i!}$ with respect to θ:

$$\ln L(\theta) = -2n\theta + \sum y_i \ln(2\theta) - \ln(\prod y_i!).$$

Taking the derivate and setting it equal to zero yields:

$$\frac{d}{d\theta} \ln L(\theta) = -2n + 2\frac{\sum y_i}{2\theta} \overset{set}{=} 0 \quad \Rightarrow \quad \hat{\theta}_{MLE} = \frac{\bar{Y}}{2}.$$

□

B. (3 points) Suppose reviews of asthma mortality rates around the world are rare in Western countries, with typical asthma mortality rates around 0.6 per 100,000. To account for this, we assume θ follows a prior distribution with $\theta \sim Gamma(\alpha, \beta)$. Suppose we also want to set the posterior $Var(\theta) = 0.12$. To what should the prior parameters α and β be set to reflect this information?

Solution.

Denote "typical" to mean "on average" (i.e. the mean). Thus $E(\theta) = \alpha\beta = 0.6$ and $Var(\theta) = \alpha\beta^2 = 0.12$. Combining this information yields $\alpha = 3$ and $\beta = 0.2$.

□

C. (3 points) Write down the function $q(\theta|Y = 3) = L(\theta) \times \pi(\theta)$.

Solution.

By straight froward multiplication, we obtain:

$$L(\theta)\pi(\theta) = e^{-2n\theta} \frac{(2\theta)^{\sum y_i}}{\prod y_i!} \frac{1}{\Gamma(\alpha)\beta^\alpha} \theta^{\alpha-1} e^{-\theta/\beta}.$$

Some simplification yields:

$$L(\theta)\pi(\theta) = \frac{2^{\sum y_i}}{\Gamma(\alpha)\beta^\alpha \prod y_i!} \theta^{\sum y_i + \alpha - 1} e^{-\theta(2n + \frac{1}{\beta})}.$$

It is worth noting that the posterior is distributed $GAM(\sum y_i + \alpha, \frac{1}{2n+1/\beta})$.

□

D. (5 points) Suppose you are going to use rejection sampling to simulate 10,000 i.i.d. observations from q, using the prior distribution of your proposal function. To what should M be set in the acceptance probability?

Solution.

We should set $M = L(\hat{\theta}_{MLE})$ (by our notes). Here, $\hat{\theta}_{MLE} = \bar{Y}/2 = (3/1)/2 = \frac{3}{2}$. Thus $L(3/2) = 0.2240418 \overset{set}{=} M$.

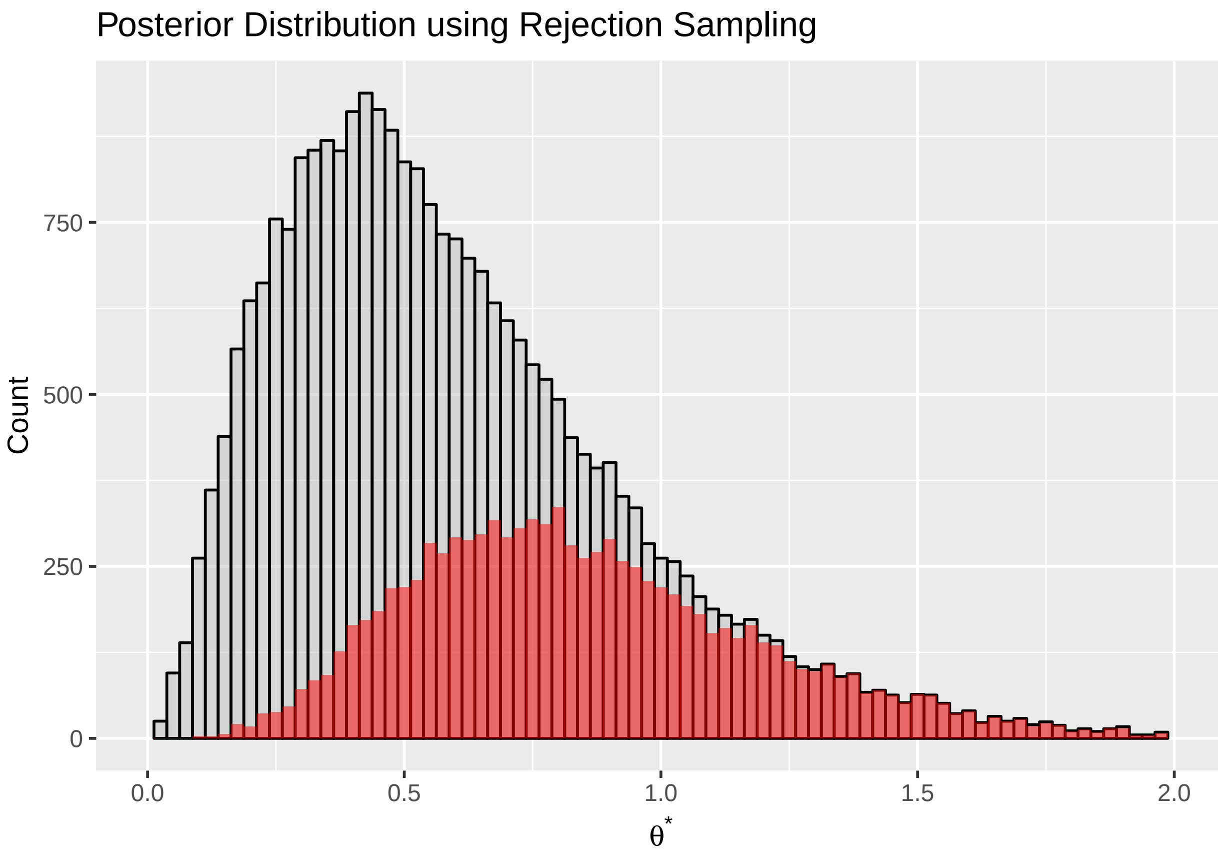
□

E. (10 points) Take a sample of size 10,000 from the posterior using rejection sampling. Include in your submission:

i. A histogram of your sample;
ii. The posterior mean;
iii. A 95% credible interval.

Solution. Consider the following code.

```
set.seed(8495)
library(ggplot2)
library(dplyr)
M <- dpois(3, lambda = 2 * (3/2))  # Liklihood evaluated at thetahat_MLE
post.samplesize <- 10000
all.thetastars <- c()
all.decisions <- c()
count <- 1
while (count <= post.samplesize) {
    thetastar <- rgamma(1, shape = 3, scale = 0.2)  # Generate 1 proposal
    accept.prob <- dpois(3, lambda = 2 * thetastar)/M
    new.decision <- rbinom(1, 1, accept.prob)
    all.thetastars <- c(all.thetastars, thetastar)
    all.decisions <- c(all.decisions, new.decision)
    count <- ifelse(new.decision == 1, count + 1, count)
}
df <- data.frame(all.thetastars, all.decisions)
posterior.sample <- df %>% filter(all.decisions == 1)
# Answer to i:
ggplot() + geom_histogram(data = df, aes(x = all.thetastars),
    binwidth = 0.025, alpha = 0.5, fill = "grey", color = "black") +
    geom_histogram(data = posterior.sample, aes(x = all.thetastars),
        binwidth = 0.025, alpha = 0.5, fill = "red") + xlab(expression(theta^"*")) +
    ylab("Count") + ggtitle("Posterior Distribution using Rejection Sampling") +
    xlim(c(0, 2))
```




```
# Answer to ii:
mean(posterior.sample$all.thetastars)
```

```
## [1] 0.86124
```

```
# Answer to iii:
quantile(posterior.sample$all.thetastars, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.3137624 1.6766739
```

□

F. (12 points) Now use the Metropolis algorithm to generate 10,000 i.i.d. observations from the posterior. (Note that 10,000 is the number of final i.i.d. observations, ***not*** the entire length of the chain, which will probably be much longer!) For full credit, you should:

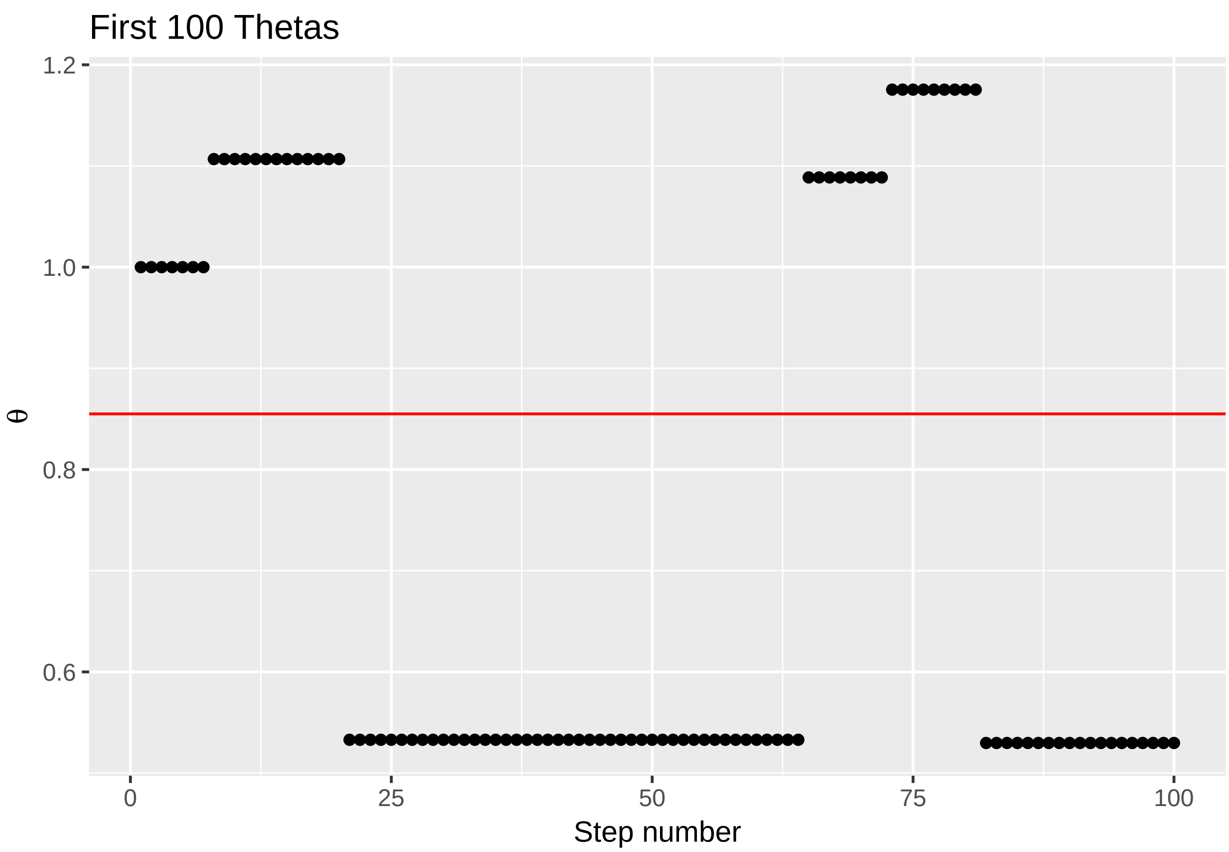
i. Discuss whether you discarded a burn-in period, and if so, how many observations you discarded (including relevant visualizations to support your decisions);
ii. Describe to what extent you thinned your Markov Chain (again using relevant visualizations to support your decisions);
iii. Create histogram, and find the posterior mean and 95% credible interval of your final (thinned, burn-in-discarded) sample.

Solution.

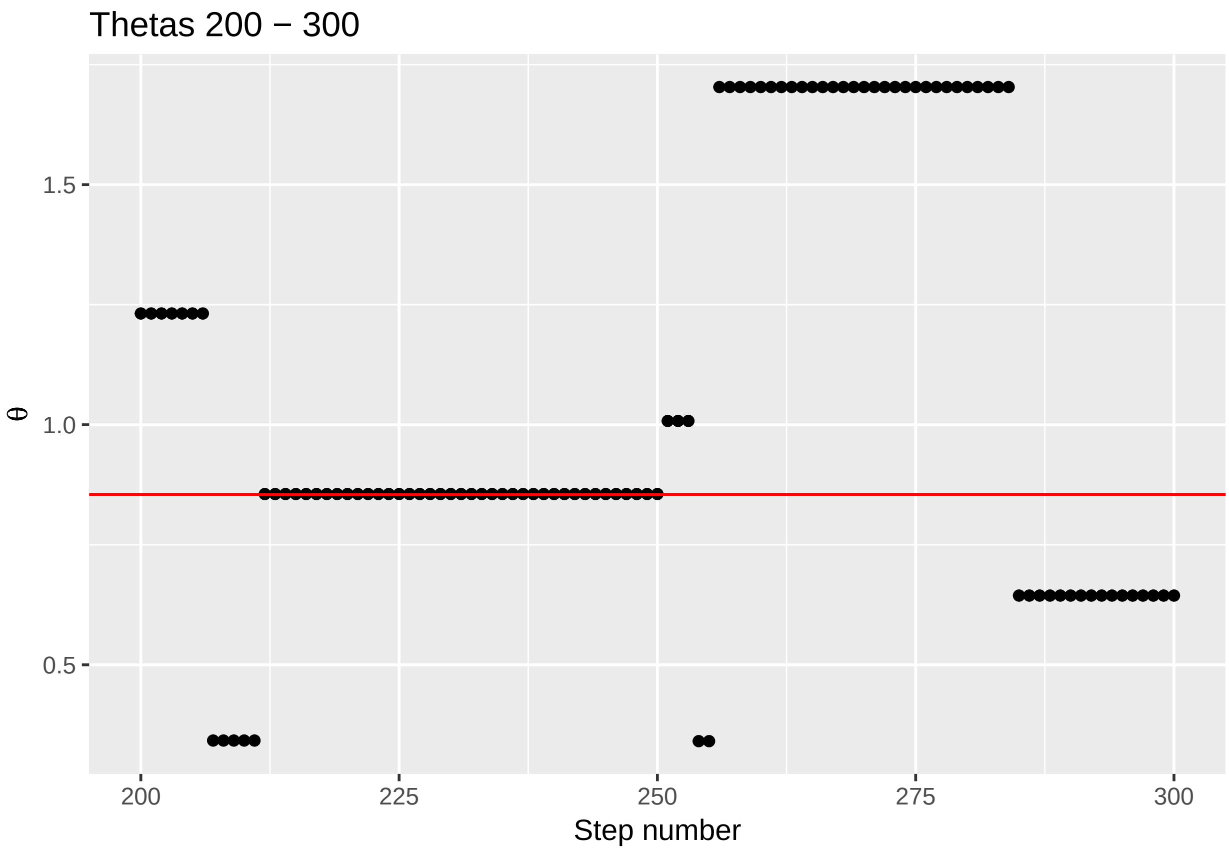
Consider the following code.

```
q.theta <- function(theta) {
    q <- ifelse(theta <= 0, 0, dpois(3, lambda = 2 * theta) *
        dgamma(theta, shape = 3, scale = 0.2))
```

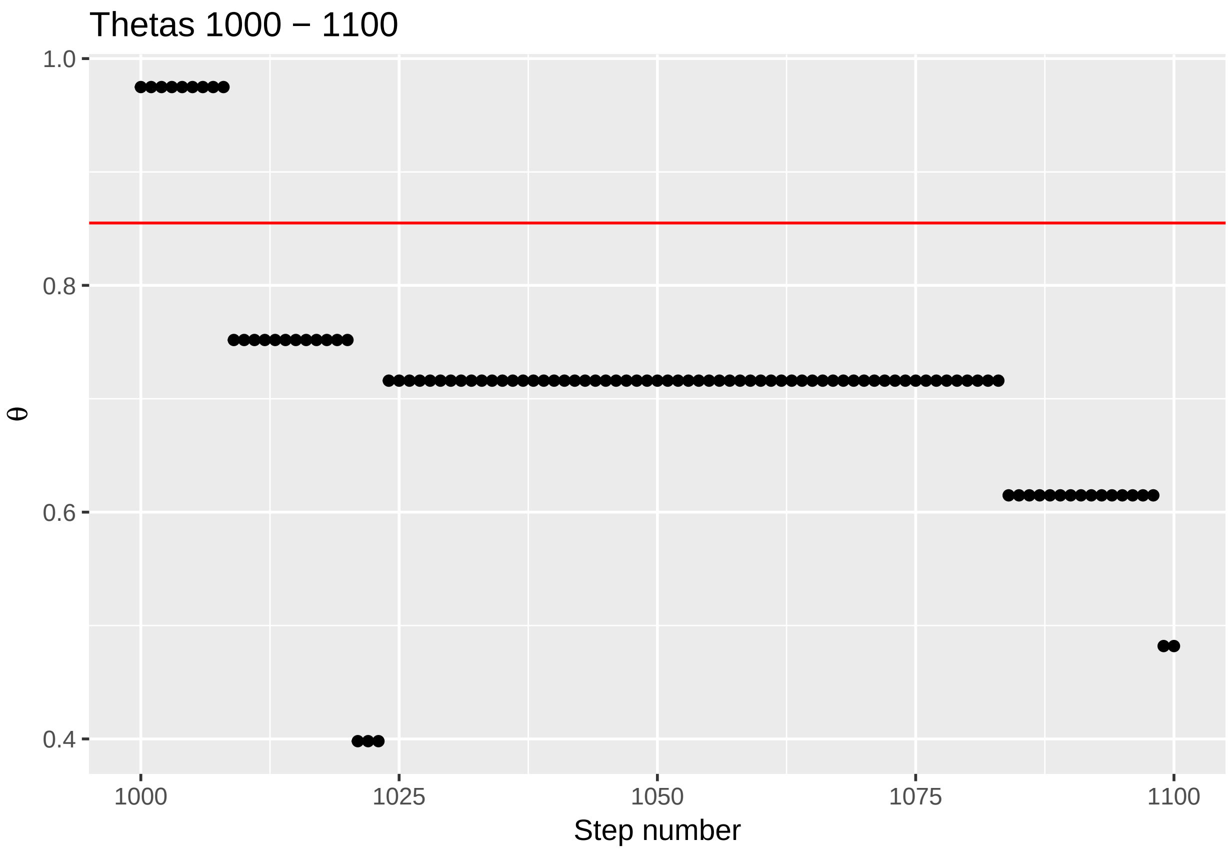
```r
    return(q)
}
post.samplesize <- 10000
thetas <- c()
thetas[1] <- 1
for (i in 2:post.samplesize) {
    thetastar <- rnorm(1, mean = 0, sd = 10)  # Proposal is NORMAL
    r <- q.theta(thetastar)/q.theta(thetas[i - 1])
    U <- runif(1)
    thetas[i] <- ifelse(U < r, thetastar, thetas[i - 1])
}
df <- data.frame(Step = 1:post.samplesize, theta = thetas)
ggplot(data = df[1:100, ]) + geom_point(aes(x = Step, y = theta)) +
    ylab(expression(theta)) + xlab("Step number") + geom_hline(aes(yintercept = 0.855),
    col = "red") + ggtitle("First 100 Thetas")
```



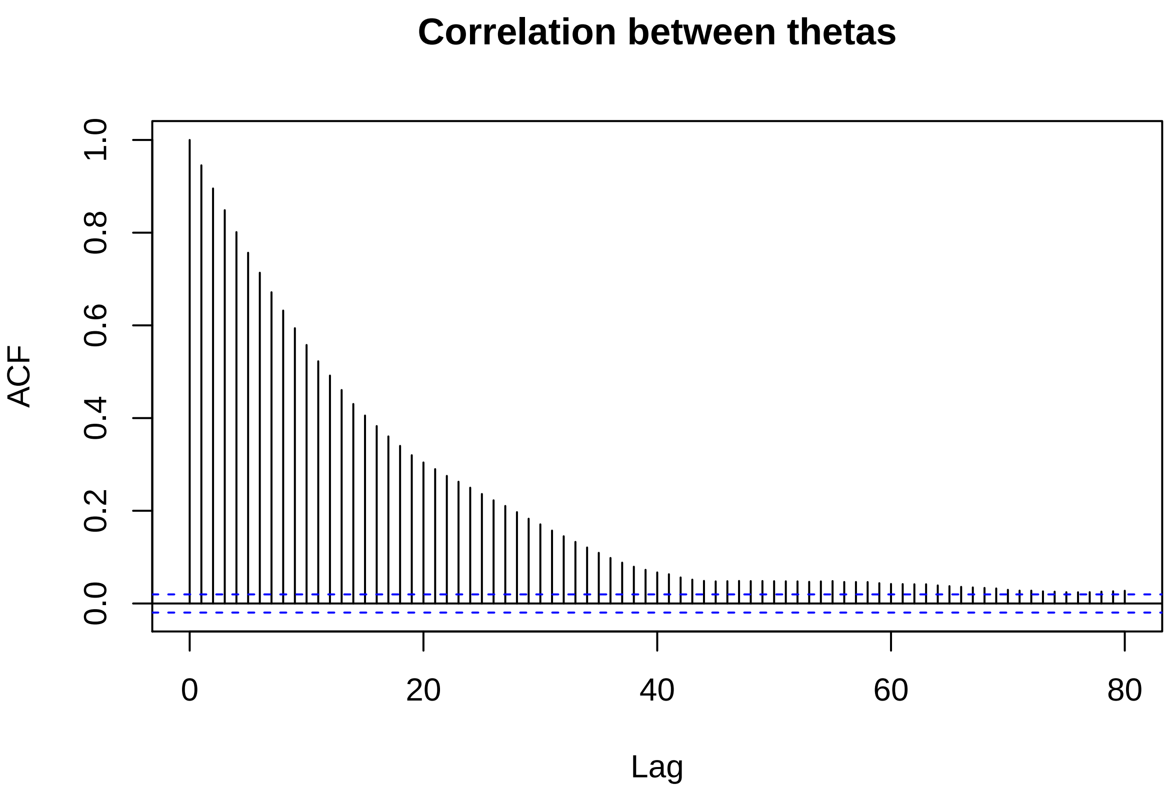

```r
ggplot(data = df[200:300, ]) + geom_point(aes(x = Step, y = theta)) +
    ylab(expression(theta)) + xlab("Step number") + geom_hline(aes(yintercept = 0.855),
    col = "red") + ggtitle("Thetas 200 - 300")
```




```
ggplot(data = df[1000:1100, ]) + geom_point(aes(x = Step, y = theta)) +
    ylab(expression(theta)) + xlab("Step number") + geom_hline(aes(yintercept = 0.855),
    col = "red") + ggtitle("Thetas 1000 - 1100")
```

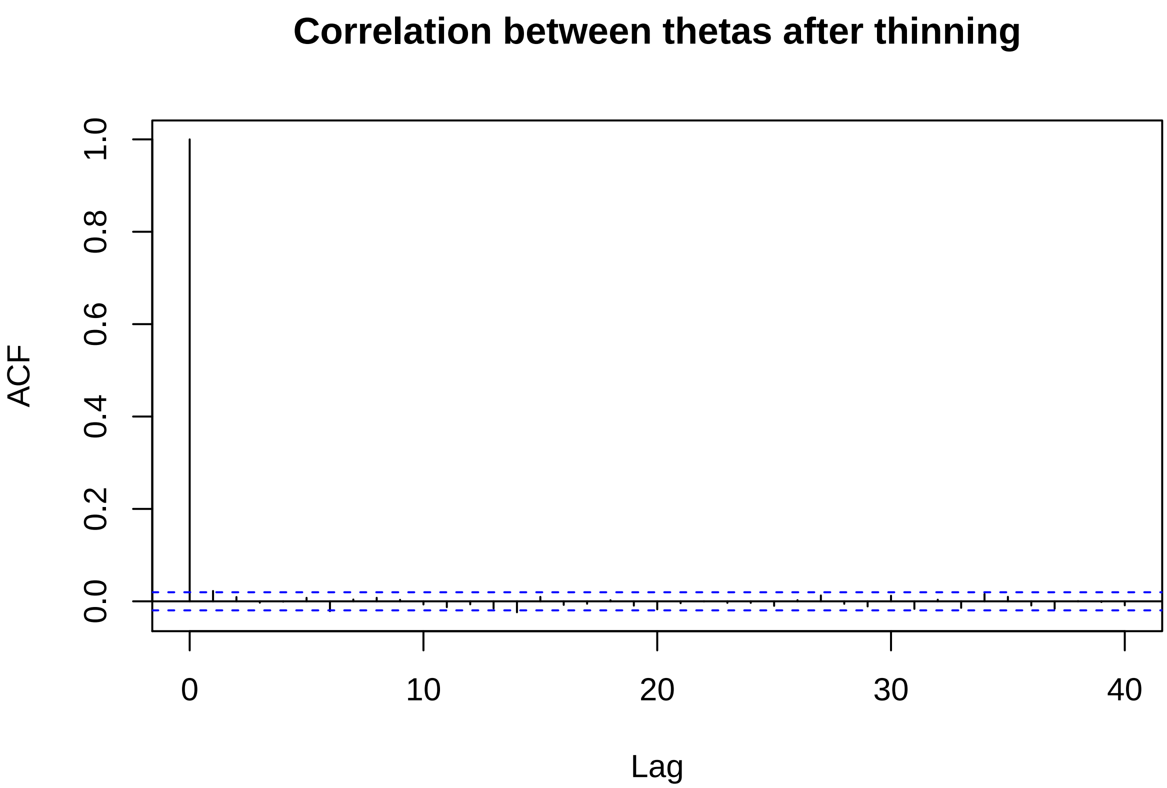



```
acf(df$theta, lag.max = 80, xlab = "Lag", ylab = "ACF", main = "Correlation between thetas")
```

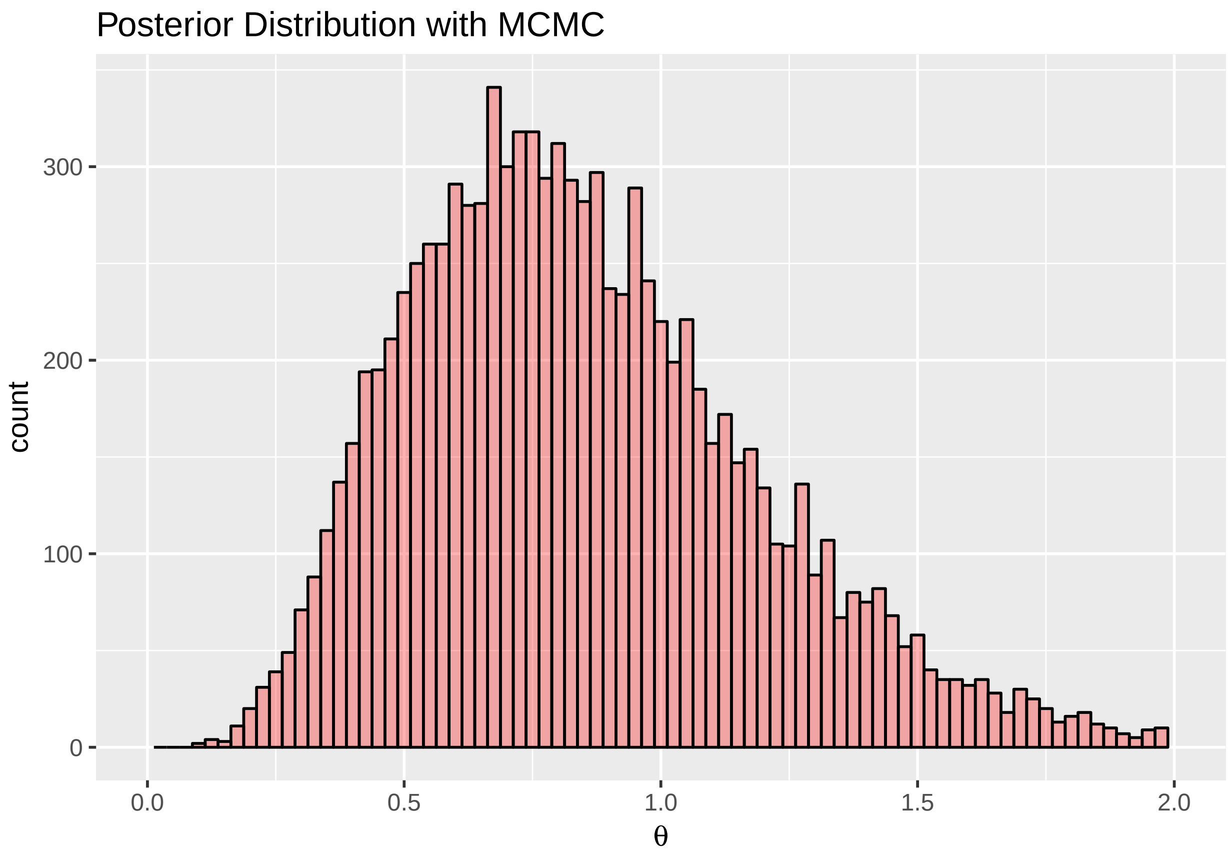


Using the above plots, we find it ideal to discard the first 1100 observations from the data set and only use every 80th realization; this will eliminate the "burn-in" period and also thin the Markov chain. Thus, the following code will create the desired posterior distribution.

```
new.post.sampsize <- 801100
thetas <- c()
thetas[1] <- 1
for (i in 2:new.post.sampsize) {
    thetastar <- rnorm(1, mean = 0, sd = 10)
    r <- q.theta(thetastar)/q.theta(thetas[i - 1])
    U <- runif(1)
    thetas[i] <- ifelse(U < r, thetastar, thetas[i - 1])
}
df <- data.frame(Step = 1:new.post.sampsize, theta = thetas)
df.burn <- df[-(1:1100), ]
df.thin <- df.burn[seq(1, nrow(df.burn), by = 80), ]
acf(df.thin$theta, xlab = "Lag", ylab = "ACF", main = "Correlation between thetas after thinning")
```




```
ggplot(data = df.thin) + geom_histogram(aes(x = theta), binwidth = 0.025,
    alpha = 0.3, fill = "red", col = "black") + ggtitle("Posterior Distribution with MCMC") +
    xlab(expression(theta)) + xlim(c(0, 2))
```




```r
mean(df.thin$theta)
```

```
## [1] 0.8544099
```

```r
quantile(df.thin$theta, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.3172745 1.6588628
```

Thus we can see by both rejection sampling and MCMC methods, $\hat{\theta}_{BAYES} \approx 0.85$.

$\square$