Homework 6, STA 360

Jerry Xin

2. Researchers are studying the length of life (lifetime) following a particular medical intervention, such as a new surgical treatment for heart disease, where the study consists of 12 patients. Specifically, the number of years before death for each is

$$3.4, 2.9, 1.2+, 1.4, 3.2, 1.8, 4.6, 1.7+, 2.0+, 1.4+, 2.8, 0.6+$$

where the + indicates that the patient was alive after x years, but the researchers lost contact with the patient after that point in time.

One way we can model this data is in the following way:

$$X_i = \begin{cases} Z_i & \text{if } Z_i \leq c_i \\ c_i & \text{if } Z_i > c_i \end{cases} \tag{1}$$

$$Z_1, \ldots, Z_n | \theta \overset{iid}{\sim} \text{Gamma}(r, \theta) \tag{2}$$

$$\theta \sim \text{Gamma}(a, b) \tag{3}$$

where a, b, and r are known. In addition, we know:

- c_i is the censoring time for patient i, which is fixed, but known only if censoring occurs.
- X_i is the observation
  - if the lifetime is less than c_i then we get to observe it ($X_i = Z_i$),
  - otherwise all we know is the lifetime is greater than c_i ($X_i = c_i$).
- θ is the parameter of interest—the rate parameter for the lifetime distribution.
- Z_i is the lifetime for patient i, however, this is not directly observed.

The probability density function (pdf) associated consists of two point masses:one at Z_i and one at c_i. The formula is

$$p(x_i|z_i) = \mathbf{1}(x_i = z_i)\mathbf{1}(z_i \leq c_i) + \mathbf{1}(x_i = c_i)\mathbf{1}(z_i > c_i).$$

.

Now we can easily find the full conditionals (derived in class and reproduced below). Notice that z_i is conditionally independent of z_j given θ for $i \neq j$. This implies that x_i is conditionally independent of x_j given z_i for $i \neq j$. Now we have

$$\begin{aligned} p(z_i|z_{-i}, x_{1:n}, \theta) &= p(z_i|x_i, \theta) \\ &\underset{z_i}{\propto} p(z_i, x_i, \theta) \\ &= p(\theta)p(z_i|\theta)p(x_i|z_i, \theta) \\ &\underset{z_i}{\propto} p(z_i|\theta)p(x_i|z_i, \theta) \\ &= p(z_i|\theta)p(x_i|z_i). \end{aligned}$$

There are now two cases to consider. If $x_i \neq c_i$, then $p(z_i|\theta)p(x_i|z_i)$ is only non-zero when $z_i = x_i$. The density devolves to a point mass at x_i. This corresponds to the case where z_i is observed, so x_i is the observed value and we should always sample this value. Practically speaking, we do not sample this value when running the Gibbs sampler.

If $x_i = c_i$, then the density becomes $p(x_i|z_i) = \mathbf{1}(z_i > c_i)$, so

$$p(z_i|\ldots) \propto p(z_i|\theta)\mathbf{1}(z_i > c_i),$$

which is a truncated Gamma.

For the Gibbs sampler, we will use the current value of θ to impute the censored data. We will sample from the truncated gamma using a modified version of the iverse CDF trick. For the censored values of Z_i we know c_i. If we know θ (which we will in a Gibbs' sampler), we know the distribution of $Z_i|\theta \sim Gamma(r, \theta)$. Let F be the CDF of this distribution. Suppose we truncate this distribution to (c, ∞). The new CDF is

$$P(Z_i < z) = \frac{F(z) - F(c)}{1 - F(c)}.$$

Therefore Y is a sample from the truncated Gamma, as desired.

In the actual code for the Gibbs' sampler we do not sample the observed values. We simply impute the censored values using the method above.

You will find code below (that is also taken from class) that will help you with the remainder of the problem.

1. (5 points) Write code to produce trace plots and running average plots for the censored values for 200 iterations. Do these diagnostic plots suggest that you have run the sampler long enough? Explain.
2. (5 points) Now run the chain for 10,000 iterations and update your diagnostic plots (traceplots and running average plots). Report your findings for both traceplots and the running average plots for θ and the censored values. Do these diagnostic plots suggest that you have run the sampler long enough? Explain.
3. (5 points) Give plots of the estimated density of $\theta \mid \cdots$ and $z_9 \mid \cdots$. Be sure to give brief explanations of your results and findings. (Present plots for 10,000 iterations).
4. (5 points) Finally, let's suppose that $r = 10, a = 1, b = 100$. Do the posterior densities in part (c) change for $\theta \mid \cdots$ and $z_9 \mid \cdots$? Do the associated posterior densities change when $r = 10, a = 100, b = 1$? Please provide plots and an explanation to back up your answer. (Use 10,000 iterations for the Gibbs sampler).

2(a) (5 points) Write code to produce trace plots and running average plots for the censored values for 200 iterations. Do these diagnostic plots suggest that you have run the sampler long enough? Explain.

```r
knitr::opts_chunk$set(cache=FALSE)
library(xtable)
set.seed(123)

# Samples from a truncated gamma with
# truncation (t, infty), shape a, and rate b
# Input: t,a,b
# Output: truncated Gamma(a,b)
sampleTrunGamma <- function(t, a, b){
  # This function samples from a truncated gamma with
  # truncation (t, infty), shape a, and rate b
  p0 <- pgamma(t, shape = a, rate = b)
  x <- runif(1, min = p0, max = 1)
  y <- qgamma(x, shape = a, rate = b)
  return(y)
}

# Gibbs sampler for censored data
# Inputs:
  # this function is a Gibbs sampler
  # z is the fully observe data
  # c is censored data
  # n.iter is number of iterations
  # init.theta and init.miss are initial values for sampler
  # r,a, and b are parameters
  # burnin is number of iterations to use as burnin
# Output: theta, z
sampleGibbs <- function(z, c, n.iter, init.theta, init.miss, r, a, b, burnin = 1){

  z.sum <- sum(z)
  m <- length(c)
  n <- length(z) + m
  miss.vals <- init.miss
  res <- matrix(NA, nrow = n.iter, ncol = 1 + m)
  for (i in 1:n.iter){
    var.sum <- z.sum + sum(miss.vals)
    theta <- rgamma(1, shape = a + n*r, rate = b + var.sum)
    miss.vals <- sapply(c, function(x) {sampleTrunGamma(x, r, theta)})
    res[i,] <- c(theta, miss.vals)
  }
  return(res[burnin:n.iter,])
}

# set parameter values
r <- 10
a <- 1
b <- 1
# input data
z <- c(3.4,2.9,1.4,3.2,1.8,4.6,2.8)
c <- c(1.2,1.7,2.0,1.4,0.6)


n.iter <- 200
init.theta <- 1
init.missing <- rgamma(length(c), shape = r, rate = init.theta)
# run sampler
res <- sampleGibbs(z, c, n.iter, init.theta, init.missing, r, a, b)
```

In figure 7 and 8 we see traceplots for 200 iterations of the Gibbs sampler. It is difficult to tell whether or not the sampler has failed to converge, thus, we turn to running average plots.

In figures 9 and 4 we see running average plots for 200 iterations of the Gibbs sampler, where from all of these it is clear that after 200 iterations the sampler is having mixing issues, and should be run for long to check that "it has not failed to converge."

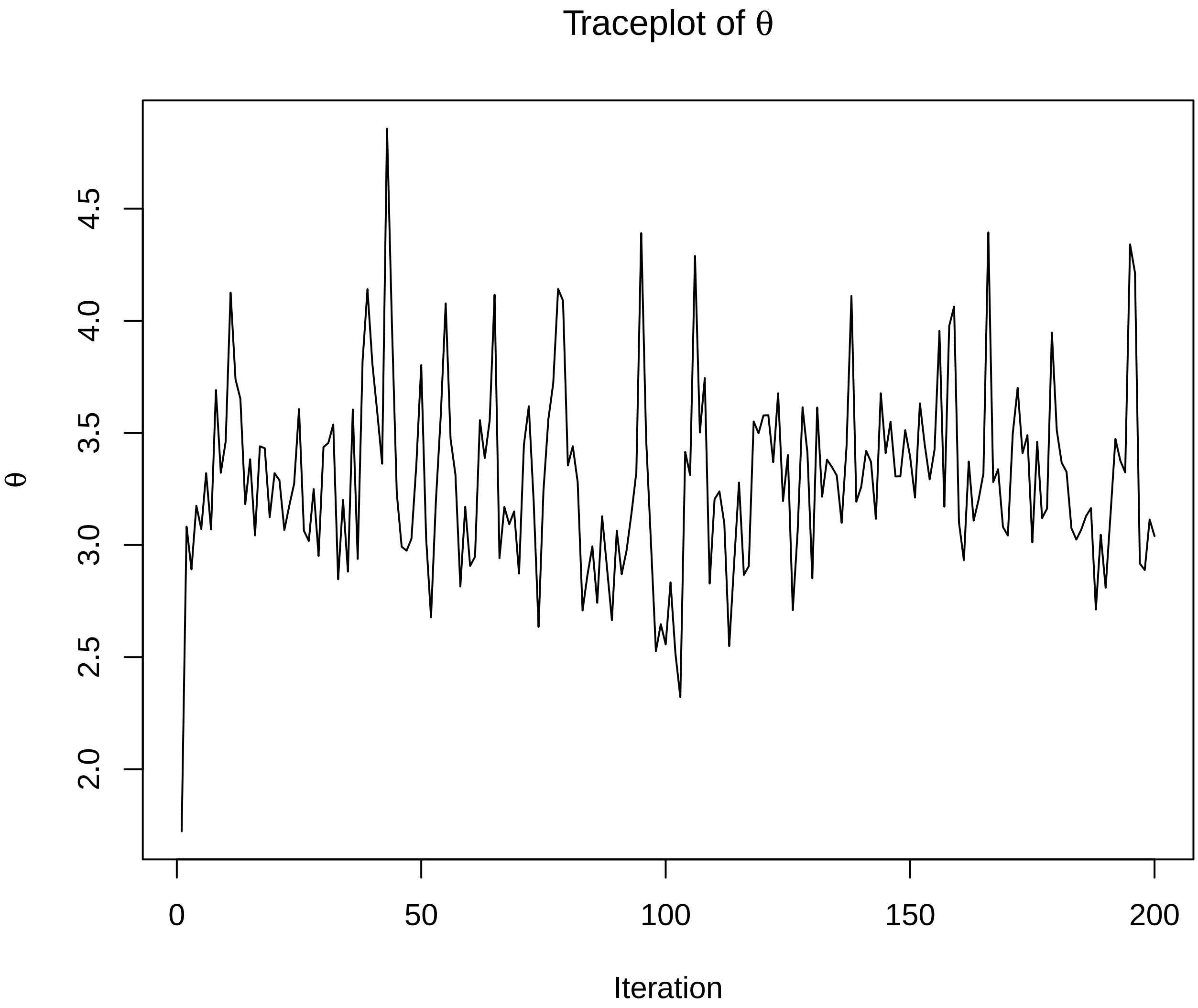


Figure 1: Traceplot of theta

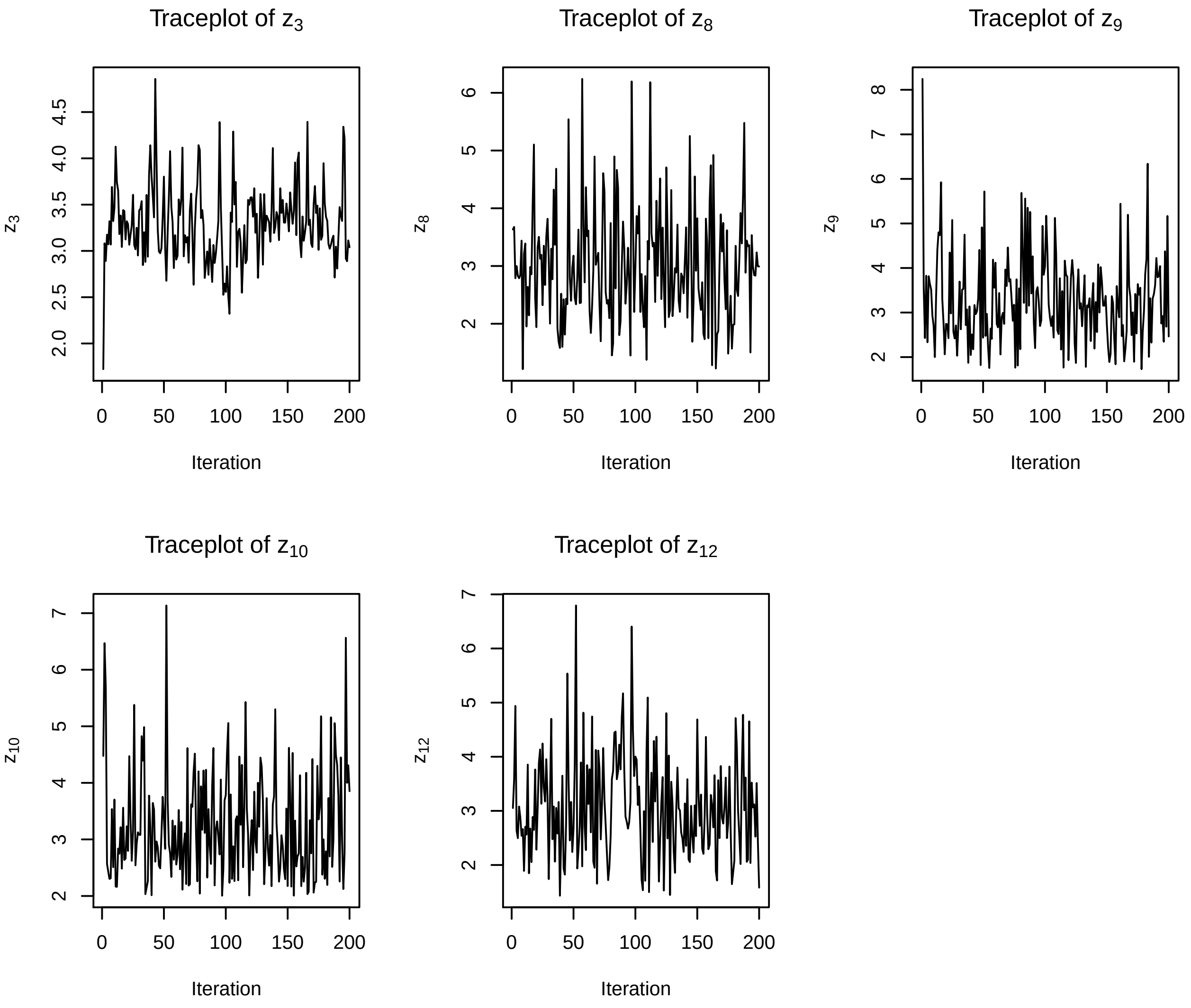


Figure 2: Traceplot of $z_3, z_8, z_9, z_{10}, z_{12}$.

```
# get running averages
set.seed(123)
run.avg <- apply(res, 2, cumsum)/(1:n.iter)
```

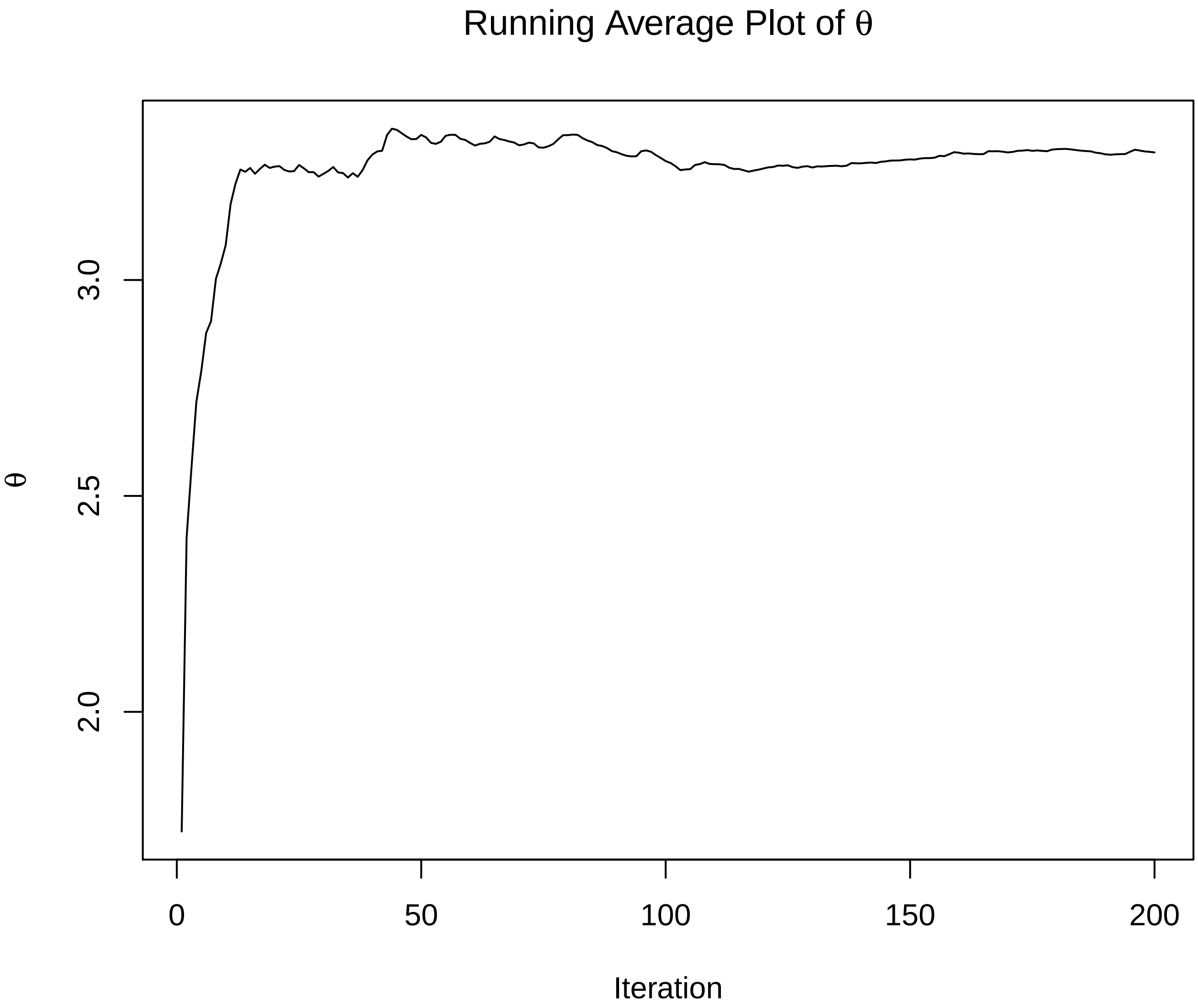


Figure 3: Running average plot of theta

Figures 13 and 12 do not provide meaniful inference at this point since the sampler has not been run long enough.

Answer to Question 2(a)

The Diagnostic Plots(Trace plots and running average plots) of 200 iterations both show that we have not run the sampler long enough, because the trace plots do not have constant variance between 0 to 200 iterations, and the running average plots have not flattened out/converged to one value and are wavering between values, between 0 to 200 iterations. It is difficult to tell whether the sample has converged or not. They seem to be beginning to converge.

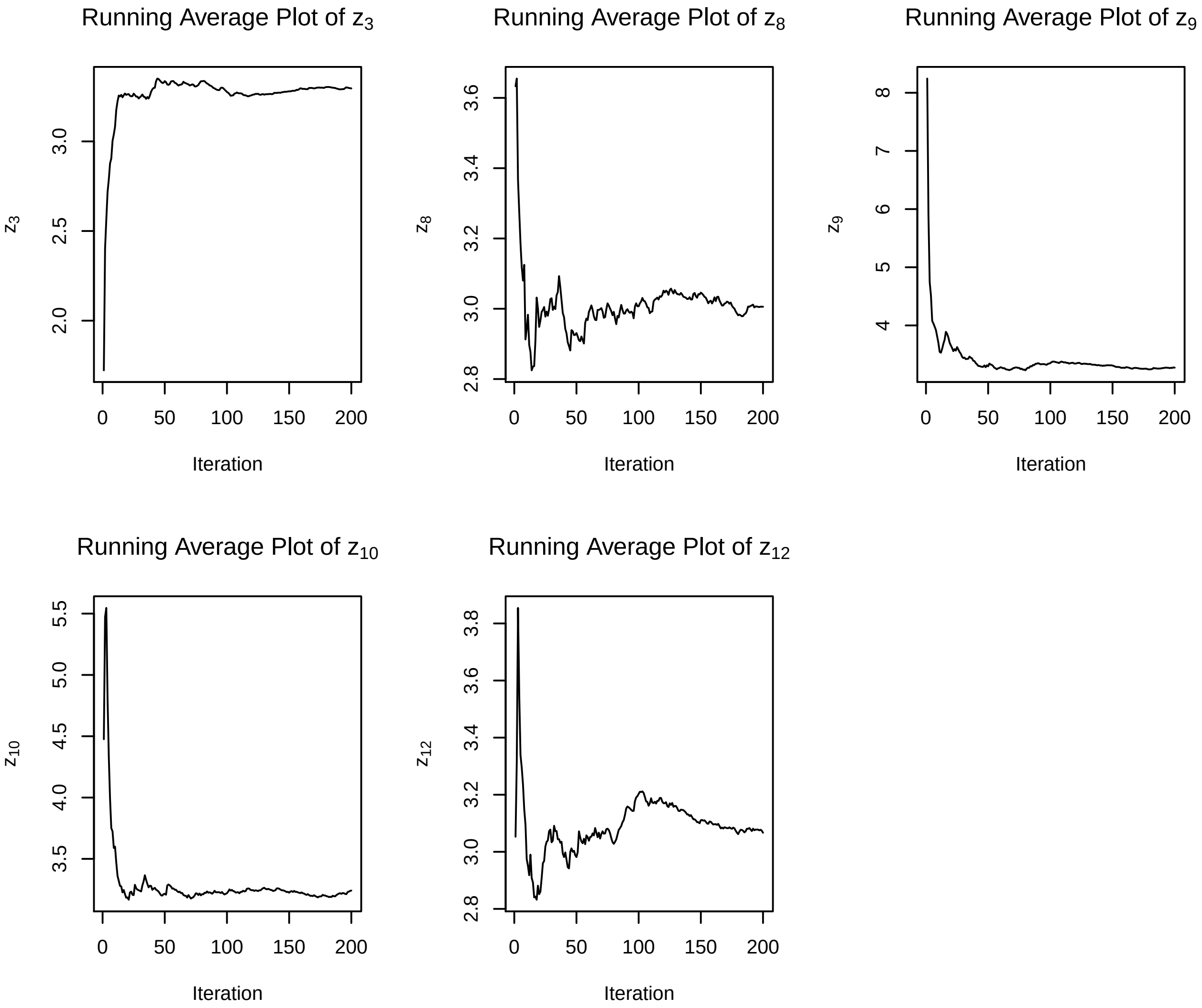


Figure 4: Running average plots of $z_3, z_8, z_9, z_{10}, z_{12}$.

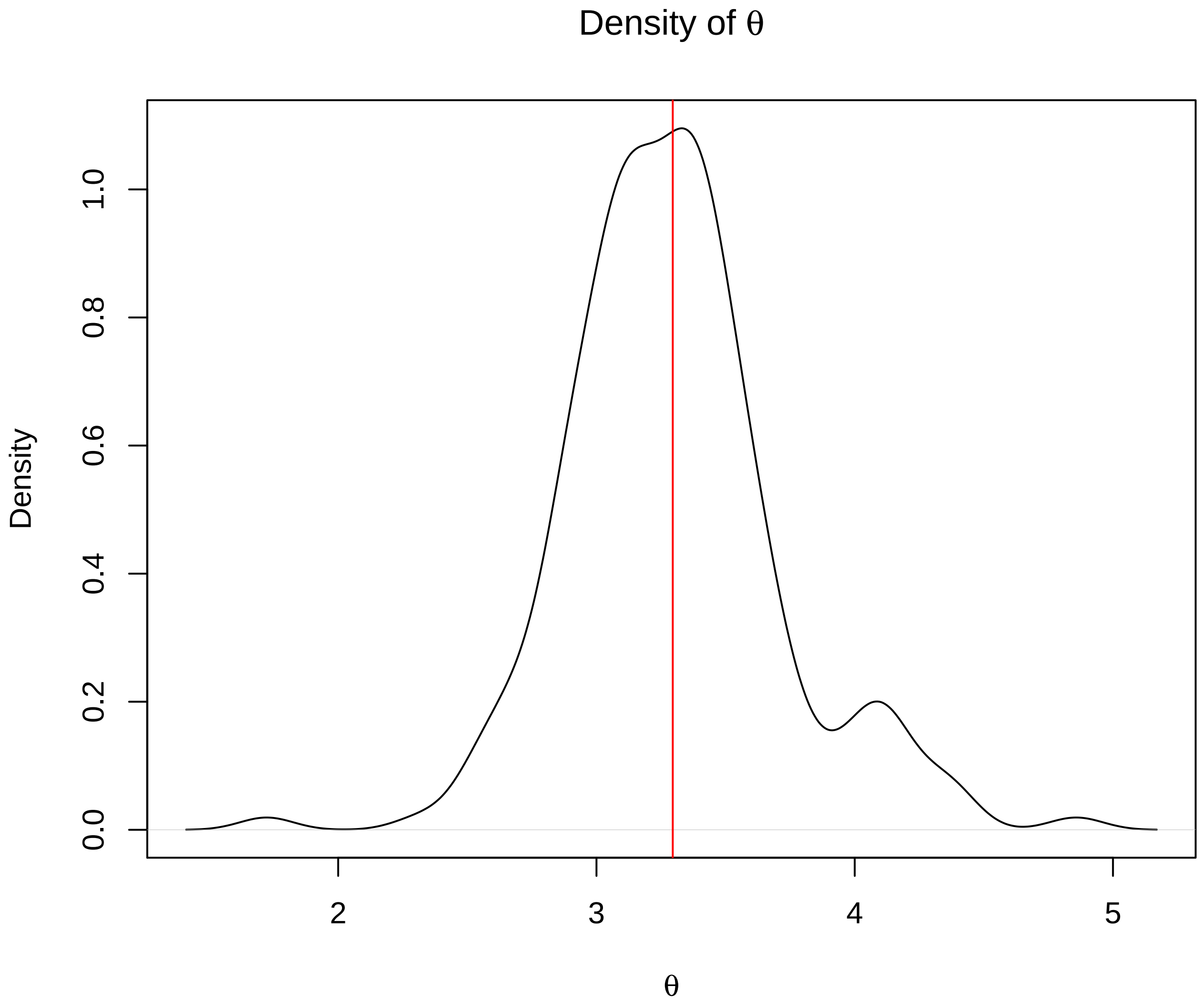


Figure 5: Estimated posterior density of theta

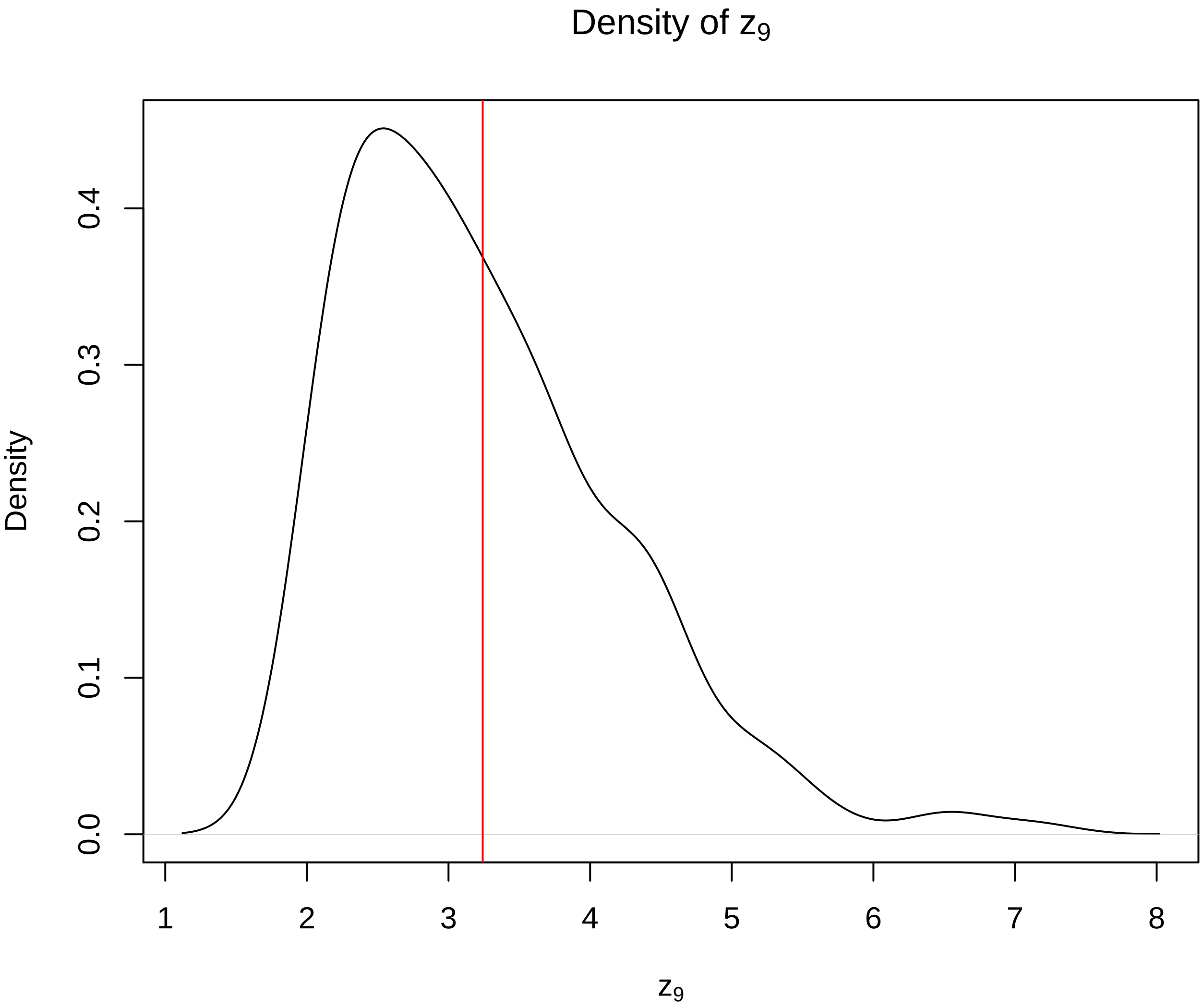


Figure 6: Estimated posterior density of z_9 (posterior mean in red).

2(b) (5 points) Now run the chain for 10,000 iterations and update your diagnostic plots (traceplots and running average plots). Report your findings for both traceplots and the running average plots for θ and the censored values. Do these diagnostic plots suggest that you have run the sampler long enough? Explain.

```
set.seed(123)
n.iter2 <- 10000
init.theta <- 1
init.missing <- rgamma(length(c), shape = r, rate = init.theta)
# run sampler
res2 <- sampleGibbs(z, c, n.iter2, init.theta, init.missing, r, a, b)
```

In figure 7 and 8 we see traceplots for 10,000 iterations of the Gibbs sampler. It is difficult to tell whether or not the sampler has failed to converge, thus, we turn to running average plots.

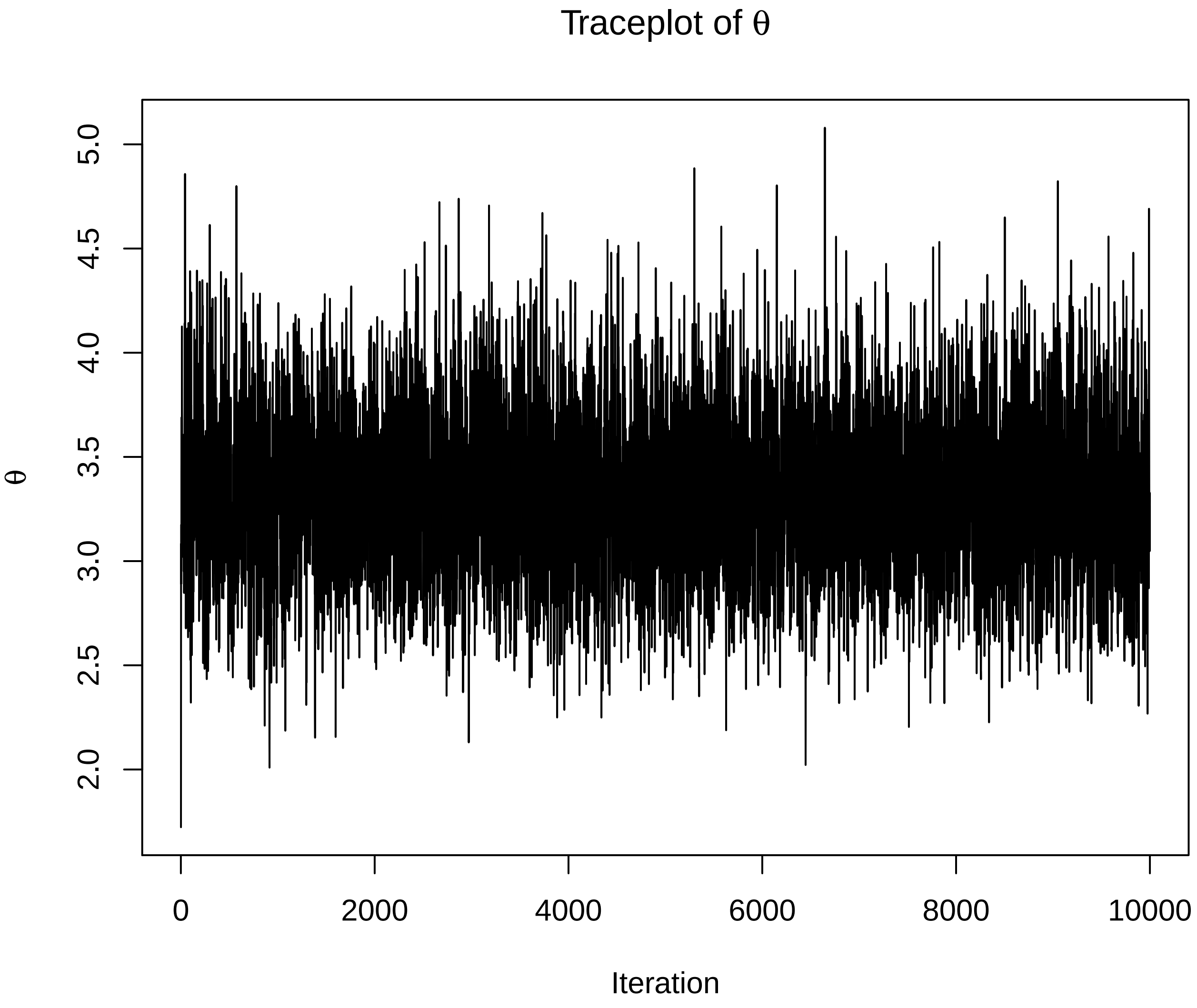


Figure 7: Traceplot of theta

In figures 9 and 4 we see running average plots for 10,000 iterations of the Gibbs sampler, where from all of these it is clear that after 10,000 iterations the sampler is having mixing issues, and should be run for long to check that “it has not failed to converge.”

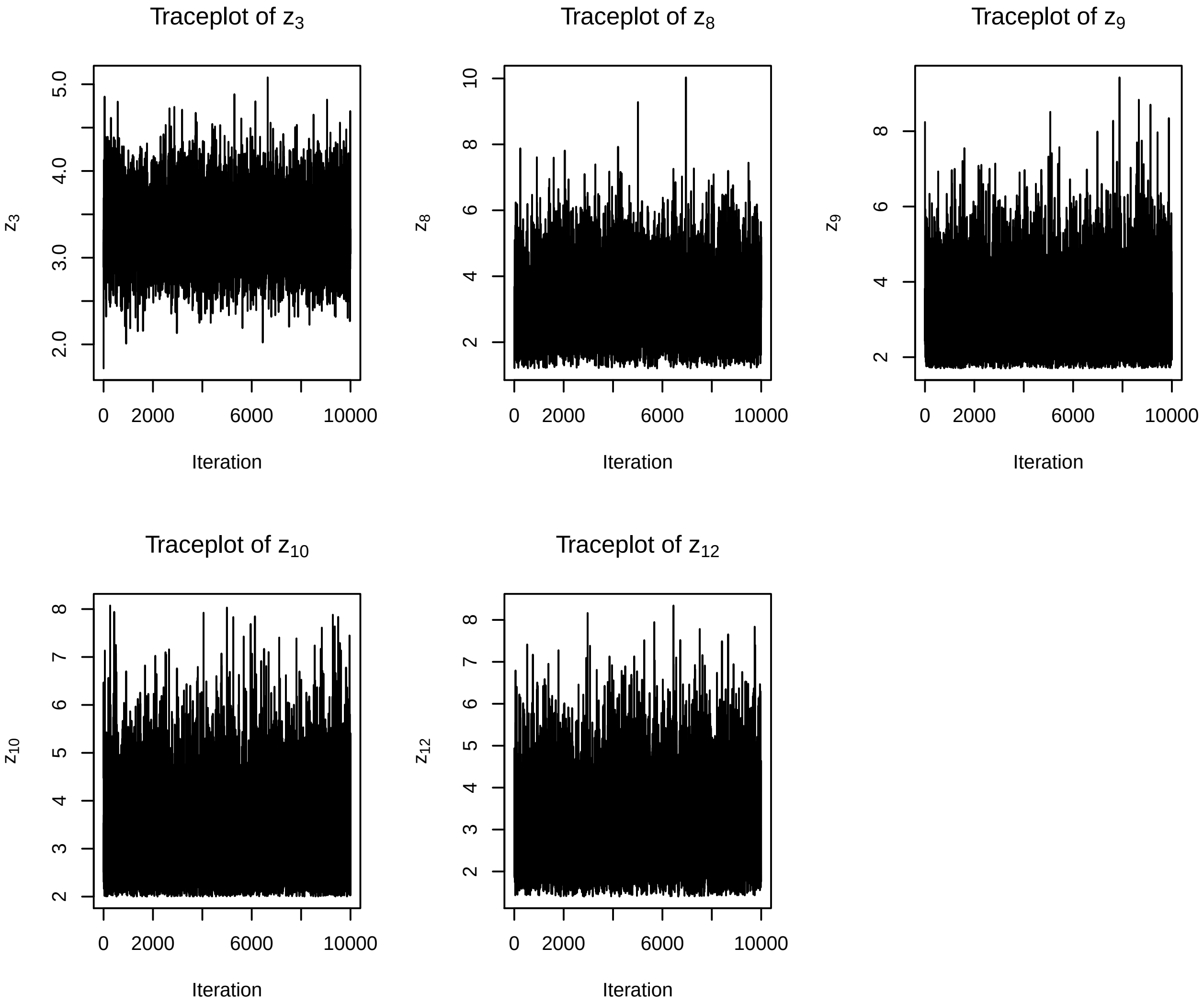


Figure 8: Traceplot of $z_3, z_8, z_9, z_{10}, z_{12}$.

Running Average for 10,000 iterations of Gibbs Sampler.

```r
# get running averages
set.seed(123)
run.avg <- apply(res2, 2, cumsum)/(1:n.iter2)
```

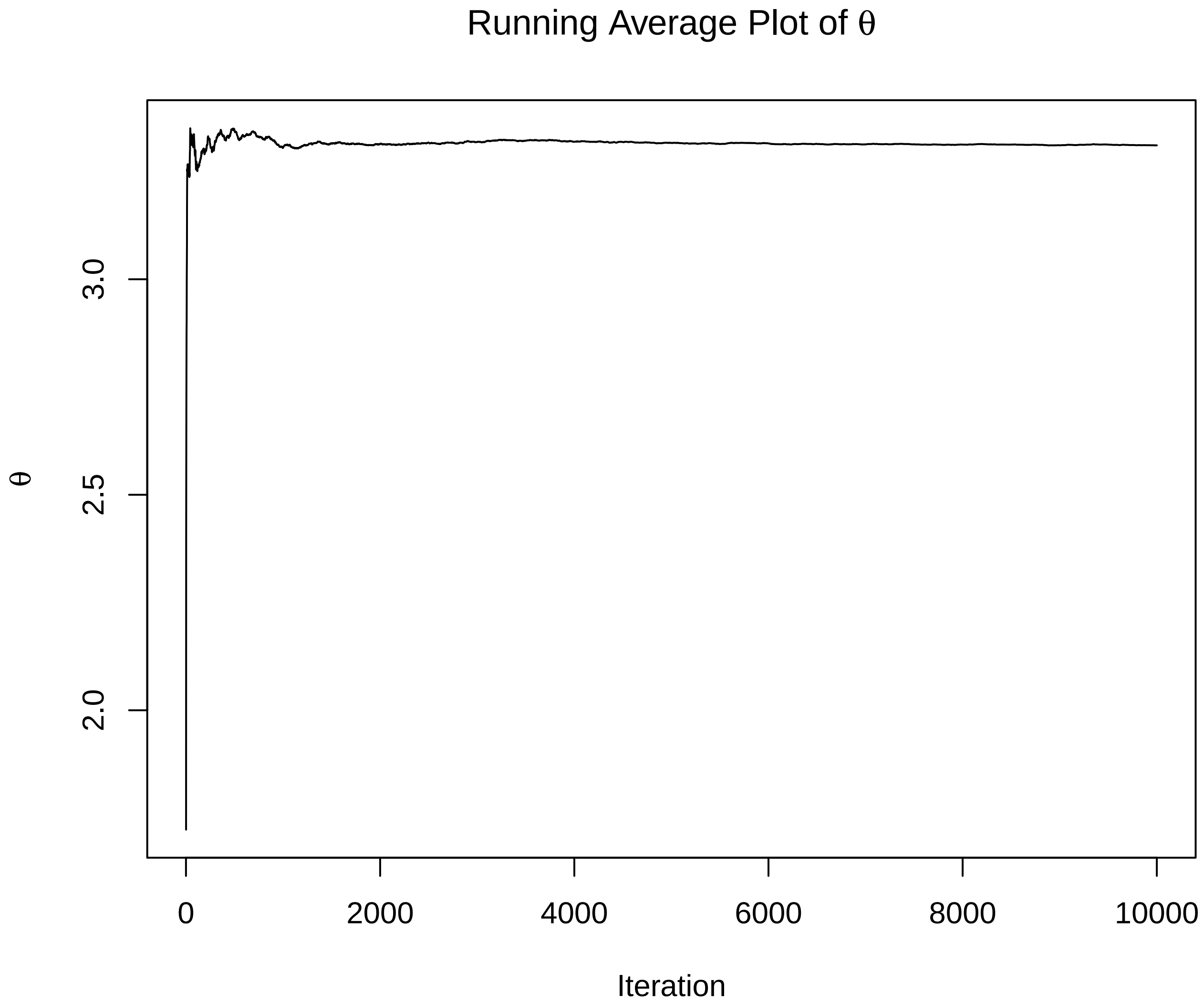
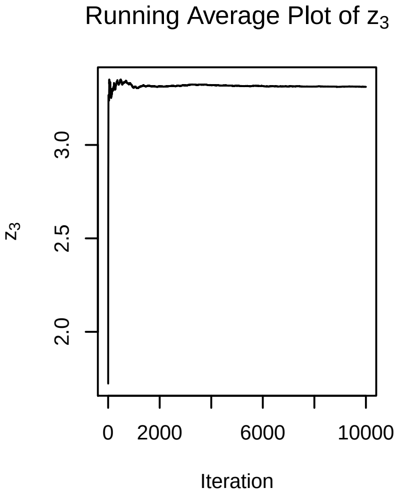
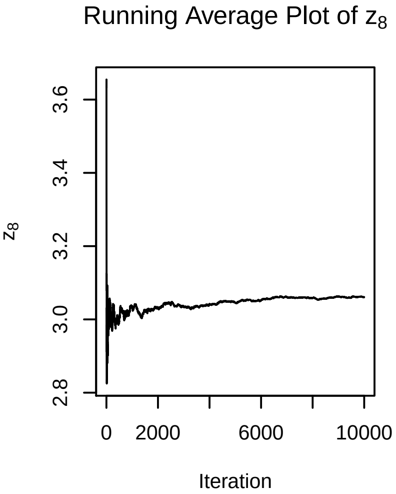
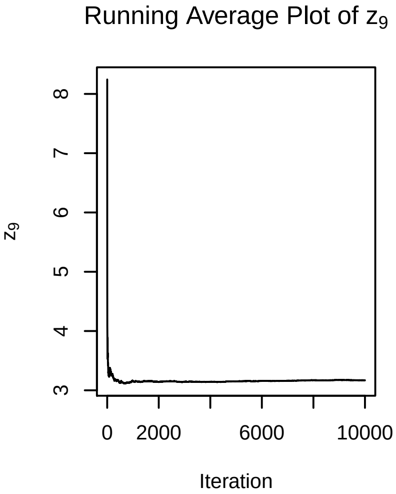
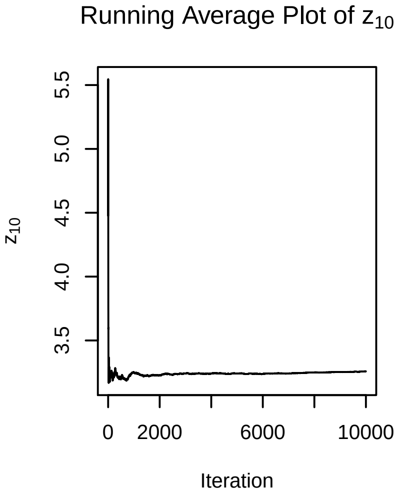


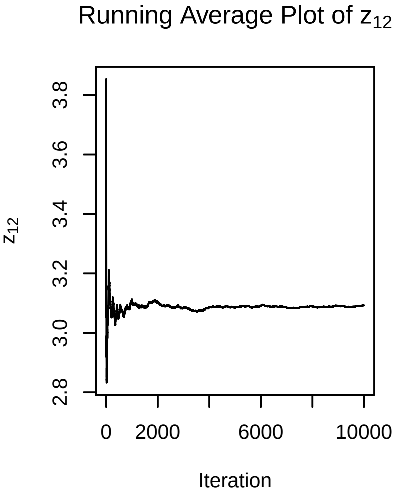
Figure 9: Running average plot of theta











## Answer to Question 2(b)

The Diagnostic Plots(Trace plots and running average plots) of 10,000 iterations both show that we have run the sampler long enough, because the trace plots have relatively constant variance between 0 to 10,000 iterations, and the running average plots have flattened out/converged to one value and are not wavering between values, between 0 to 10,000 iterations. There seems to be no evidence that our sample fails to converge. The running average plot of theta seems to have converged to around 3.3.

#2(c) (5 points) Give plots of the estimated density of $\theta \mid \cdots$ and $z_9 \mid \cdots$. Be sure to give brief explanations of your results and findings. (Present plots for 10,000 iterations).

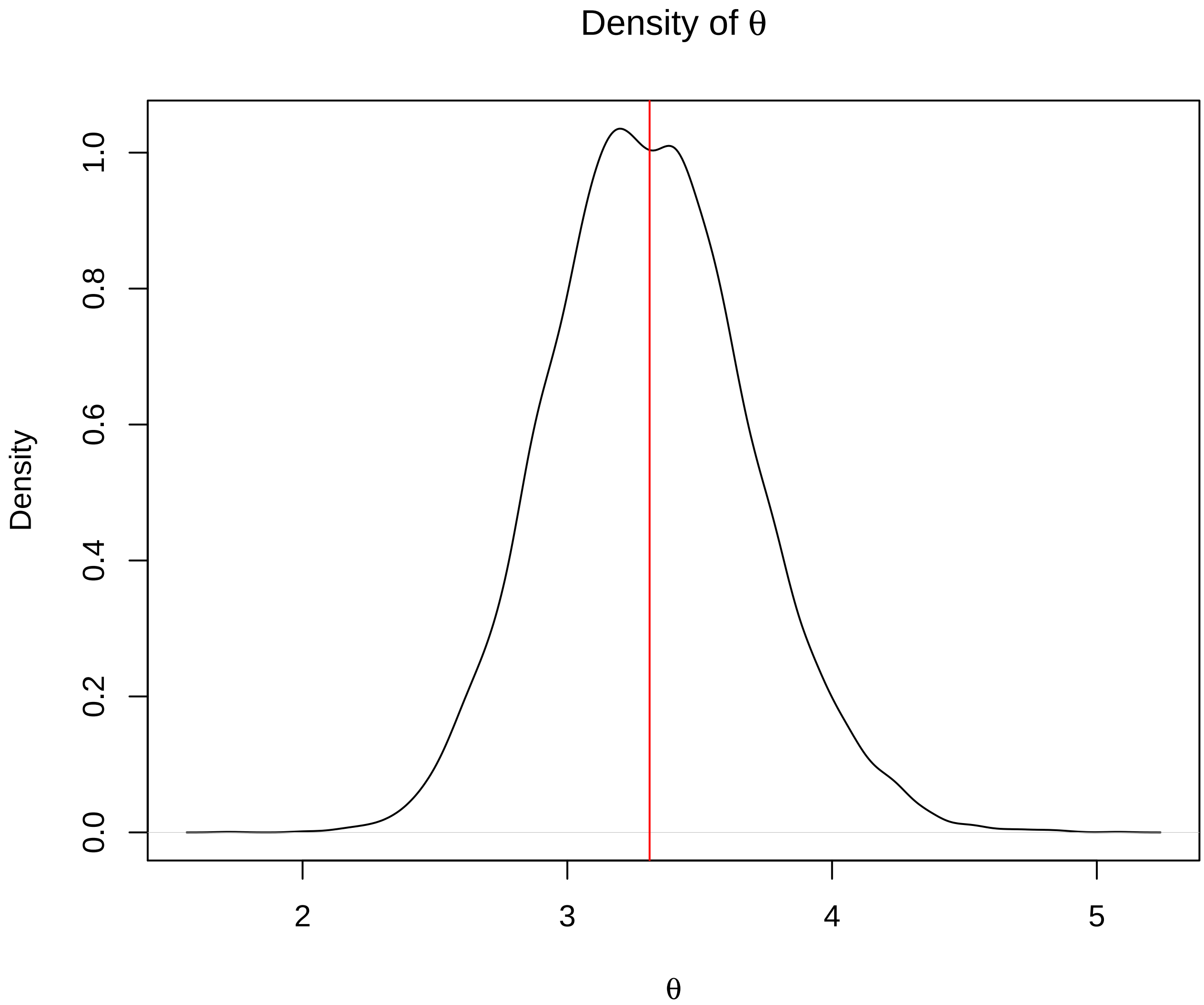
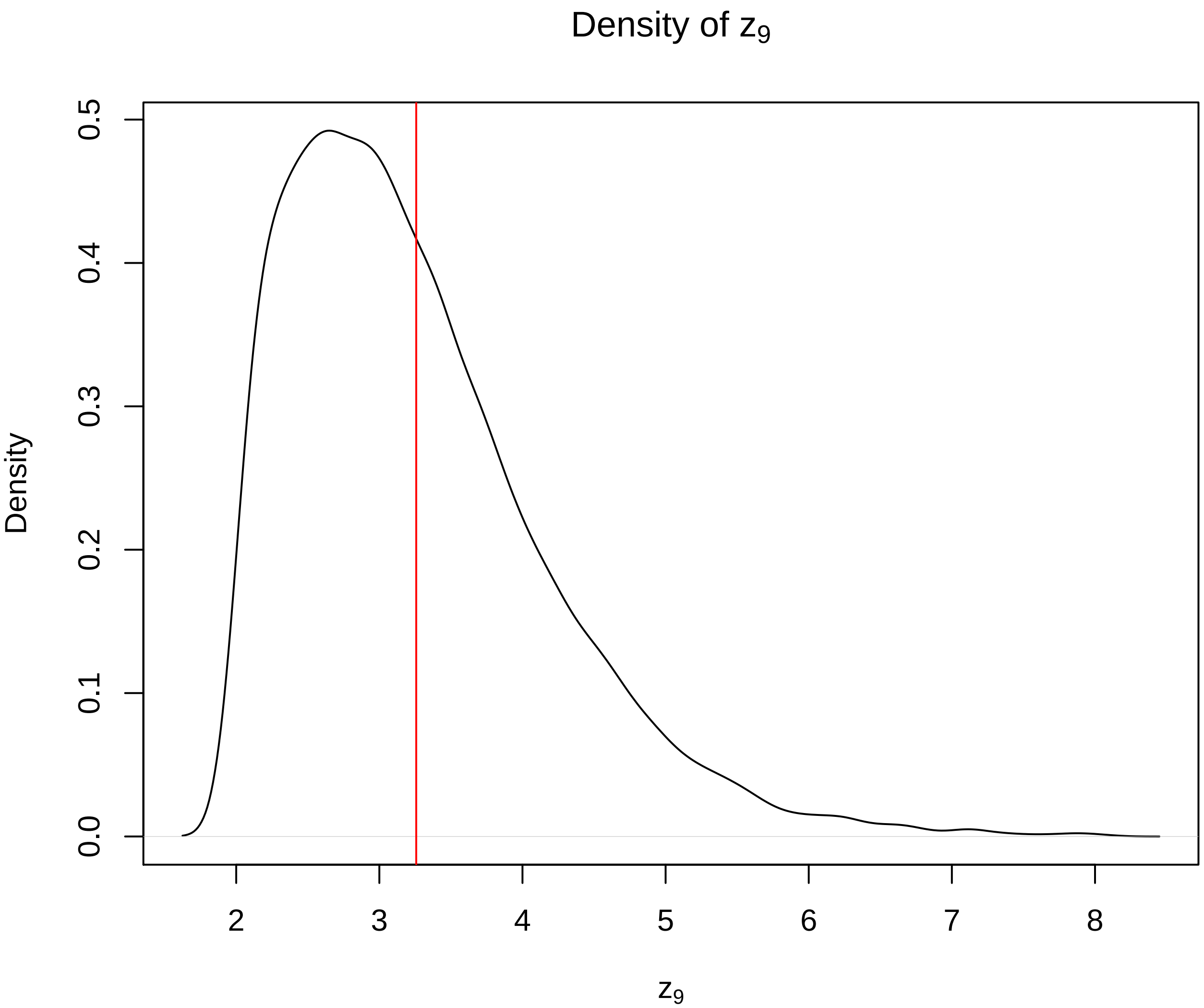


Figure 10: Estimated posterior density of theta



## Answer to Question 2(c) The Density plot of theta seems to have a posterior mean of 3.3 and is relatively smooth, which means it has converged and we have run the sampler long enough and we have a large enough sample size. The Density plot of z9 seems to have a posterior mean of 3.3 and is relatively smooth, which means it has converged and we have run the sampler long enough and we have a large enough sample size.

The estimated true value of the theta and z9 lifespans is larger than all of the observed data, which confirms our assumption that censoring is present here. We believe that the values of our observed data are artifically low, so the model is predicting comparatively large lifespan. The red line is larger because of our censoring assumption. The data seems to be aritifically small, and 3.3 is the most likely value for the true lifespan.

2(d) (5 points) Finally, let's suppose that $r = 10, a = 1, b = 100$. Do the posterior densities in part (c) change for $\theta \mid \cdots$ and $z_9 \mid \cdots$? Do the associated posterior densities change when $r = 10, a = 100, b = 1$? Please provide plots and an explanation to back up your answer. (Use 10,000 iterations for the Gibbs sampler).

```
set.seed(123)
knitr::opts_chunk$set(cache=FALSE)
library(xtable)
set.seed(123)

# Samples from a truncated gamma with
```

```r
# truncation (t, infty), shape a, and rate b
# Input: t,a,b
# Output: truncated Gamma(a,b)
sampleTrunGamma <- function(t, a, b){
  # This function samples from a truncated gamma with
  # truncation (t, infty), shape a, and rate b
  p0 <- pgamma(t, shape = a, rate = b)
  x <- runif(1, min = p0, max = 1)
  y <- qgamma(x, shape = a, rate = b)
  return(y)
}

# Gibbs sampler for censored data
# Inputs:
  # this function is a Gibbs sampler
  # z is the fully observe data
  # c is censored data
  # n.iter is number of iterations
  # init.theta and init.miss are initial values for sampler
  # r,a, and b are parameters
  # burnin is number of iterations to use as burnin
# Output: theta, z
sampleGibbs <- function(z, c, n.iter, init.theta, init.miss, r, a, b, burnin = 1){

  z.sum <- sum(z)
  m <- length(c)
  n <- length(z) + m
  miss.vals <- init.miss
  res <- matrix(NA, nrow = n.iter, ncol = 1 + m)
  for (i in 1:n.iter){
    var.sum <- z.sum + sum(miss.vals)
    theta <- rgamma(1, shape = a + n*r, rate = b + var.sum)
    miss.vals <- sapply(c, function(x) {sampleTrunGamma(x, r, theta)})
    res[i,] <- c(theta, miss.vals)
  }
  return(res[burnin:n.iter,])
}

# set parameter values
r <- 10
a <- 1
b <- 100
# input data
z <- c(3.4,2.9,1.4,3.2,1.8,4.6,2.8)
c <- c(1.2,1.7,2.0,1.4,0.6)


n.iter <- 10000
init.theta <- 1
init.missing <- rgamma(length(c), shape = r, rate = init.theta)
# run sampler
res <- sampleGibbs(z, c, n.iter, init.theta, init.missing, r, a, b)
```

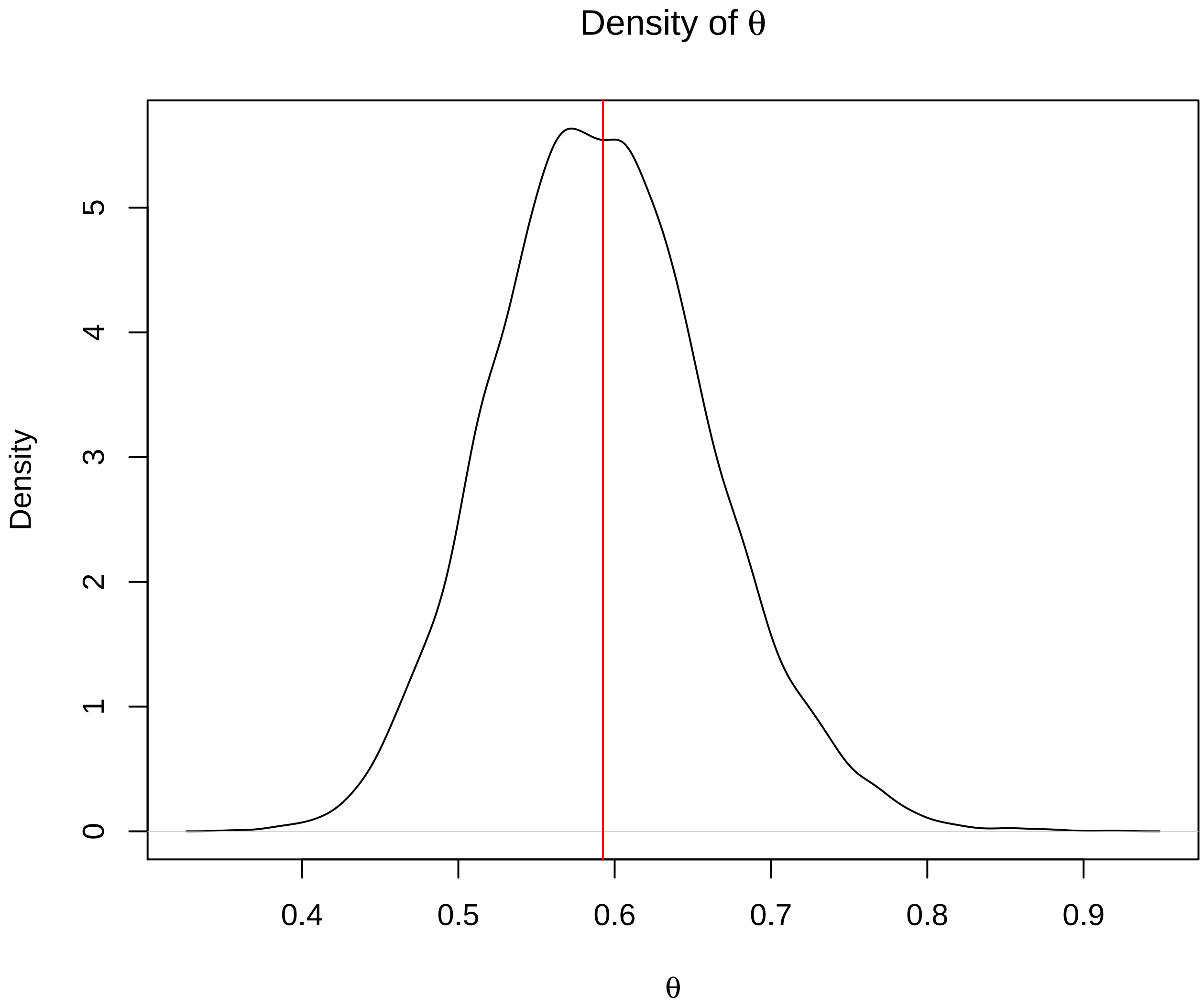


Figure 11: Estimated posterior density of theta

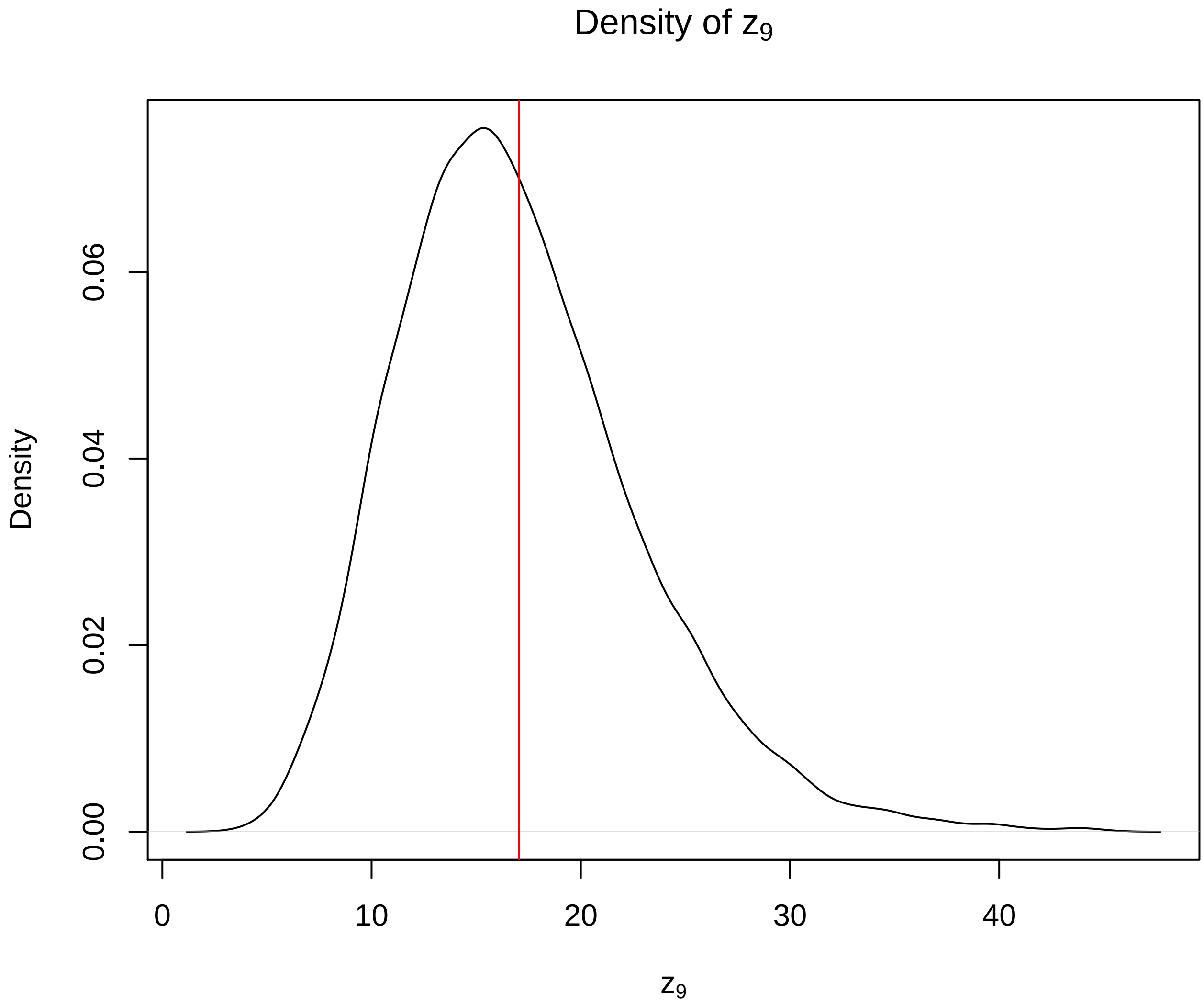


Figure 12: Estimated posterior density of z_9 (posterior mean in red).

```
set.seed(123)
knitr::opts_chunk$set(cache=FALSE)
library(xtable)
set.seed(123)

# Samples from a truncated gamma with
# truncation (t, infty), shape a, and rate b
# Input: t,a,b
# Output: truncated Gamma(a,b)
sampleTrunGamma <- function(t, a, b){
  # This function samples from a truncated gamma with
  # truncation (t, infty), shape a, and rate b
  p0 <- pgamma(t, shape = a, rate = b)
  x <- runif(1, min = p0, max = 1)
  y <- qgamma(x, shape = a, rate = b)
  return(y)
}

# Gibbs sampler for censored data
# Inputs:
  # this function is a Gibbs sampler
  # z is the fully observe data
  # c is censored data
  # n.iter is number of iterations
  # init.theta and init.miss are initial values for sampler
  # r,a, and b are parameters
  # burnin is number of iterations to use as burnin
# Output: theta, z
sampleGibbs <- function(z, c, n.iter, init.theta, init.miss, r, a, b, burnin = 1){

  z.sum <- sum(z)
  m <- length(c)
  n <- length(z) + m
  miss.vals <- init.miss
  res <- matrix(NA, nrow = n.iter, ncol = 1 + m)
  for (i in 1:n.iter){
    var.sum <- z.sum + sum(miss.vals)
    theta <- rgamma(1, shape = a + n*r, rate = b + var.sum)
    miss.vals <- sapply(c, function(x) {sampleTrunGamma(x, r, theta)})
    res[i,] <- c(theta, miss.vals)
  }
  return(res[burnin:n.iter,])
}

# set parameter values
r <- 10
a <- 100
b <- 1
# input data
z <- c(3.4,2.9,1.4,3.2,1.8,4.6,2.8)
c <- c(1.2,1.7,2.0,1.4,0.6)
```

```
n.iter <- 10000
init.theta <- 1
init.missing <- rgamma(length(c), shape = r, rate = init.theta)
# run sampler
res <- sampleGibbs(z, c, n.iter, init.theta, init.missing, r, a, b)
```

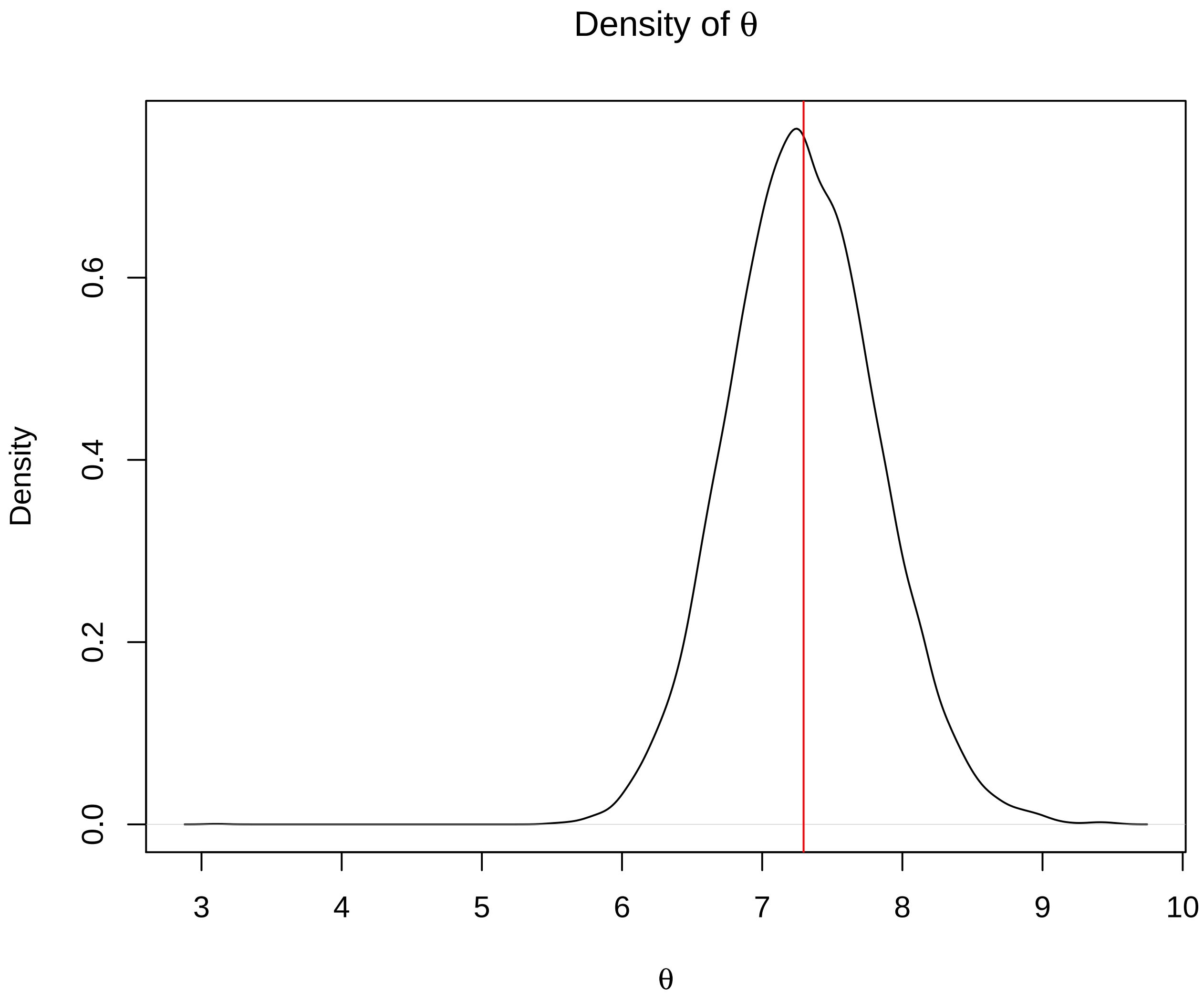
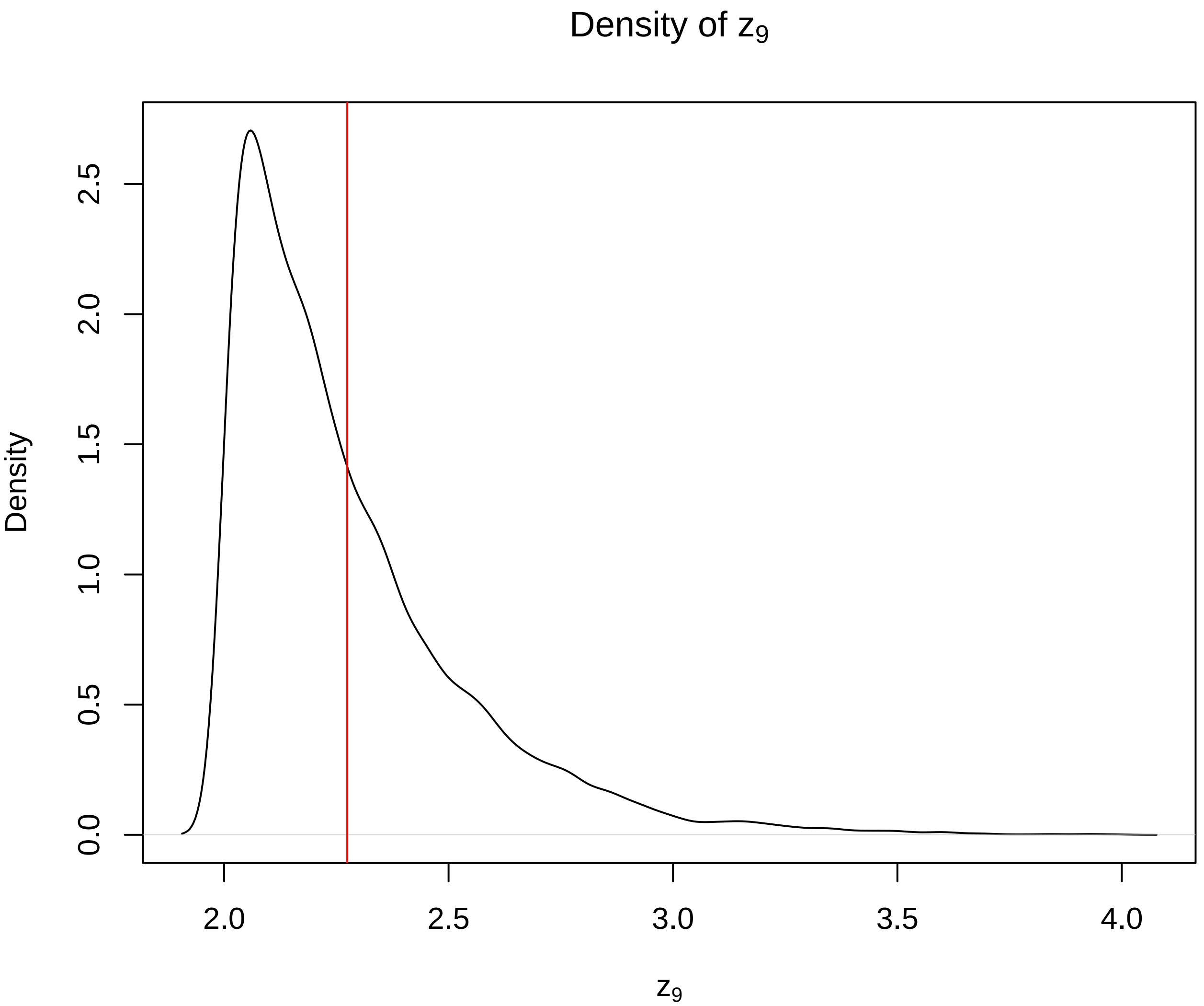


Figure 13: Estimated posterior density of theta



## Answer to Question 2(d)
Yes, the posterior densities change as you are physically changing the value of b to 100, and then a to 100. This is conceptually the same as changing the prior on lifespan. Changing the prior has a statistically significant effect on our posterior density curve, and therefore our resulting posterior.

When r = 10, a = 1, and b = 100, the Density plot of theta seems to have a posterior mean of around 0.6. It seems to have a posterior mean around 3.3 when r = 10, a = 1, b = 1. Because the posterior mean is now around 0.6 instead of 3.3, this means that our prior significantly affects the mean and width of the posterior.

When r = 10, a = 1, and b = 100, the Density plot of z9 is centered at about 17. It seems to have a posterior mean around 3.3 when r = 10, a = 1, b = 1. Because the posterior mean is now around 17 instead of 3.3, this means that our prior significantly affects the mean and width of the posterior.

When r = 10, a = 100, and b = 1, the Density plot of theta is centered at around 7.2. It seems to have a posterior mean around 3.3 when r = 10, a = 1, b = 1.Because the posterior mean is now around 7.2 instead of 3.3, this means that our prior significantly affects the mean and width of the posterior.

When r = 10, a = 100, and b = 1, the Density plot of z9 is centered at around 2.25. It seems to have a posterior mean around 3.3 when r = 10, a = 1, b = 1.Because the posterior mean is now around 2.25 instead of 3.3, this means that our prior significantly affects the mean and width of the posterior.

So, our posterior density for theta and z9 both change when we change b from 1 to 100, and when we change a from 1 to 100.