

Homework 6

STA 360: Assignment 6, Fall 2020

Due Monday October 12, 5 PM Standard Eastern Time

Question 2

Part 1

All the code below was taken from Professors Rebecca C. Steort's provided template code.

```
knitr::opts_chunk$set(cache=TRUE)
library(xtable)

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

#Create Traceplot for Theta
plot(1:n.iter, res[,1], pch = 16, cex = .35,
     xlab = "Iteration", ylab = expression(theta),
     main = expression(paste("Traceplot of ", theta, " for 200 Iterations")))

```

Traceplot of θ for 200 Iterations

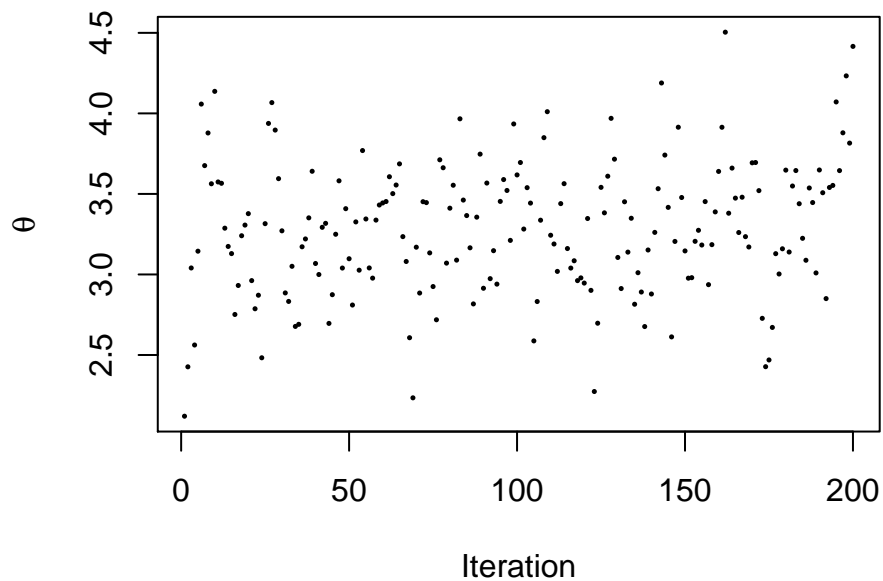


Figure 1: Traceplot of $z_3, z_8, z_9, z_{10}, z_{12}$.

```

#Create Traceplots for Censored Values
missing.index <- c(3,8,9,10,12)
par(mfrow=c(2,3))
for (ind in missing.index){
  x.lab <- bquote(z[.(ind)])
  plot(1:n.iter, res[,which(missing.index == ind)], pch = 16, cex = .35,

```

```

xlab = "Iteration", ylab = x.lab,
main = bquote(paste("Traceplot of ", .(x.lab))))
}
plot.new()

```

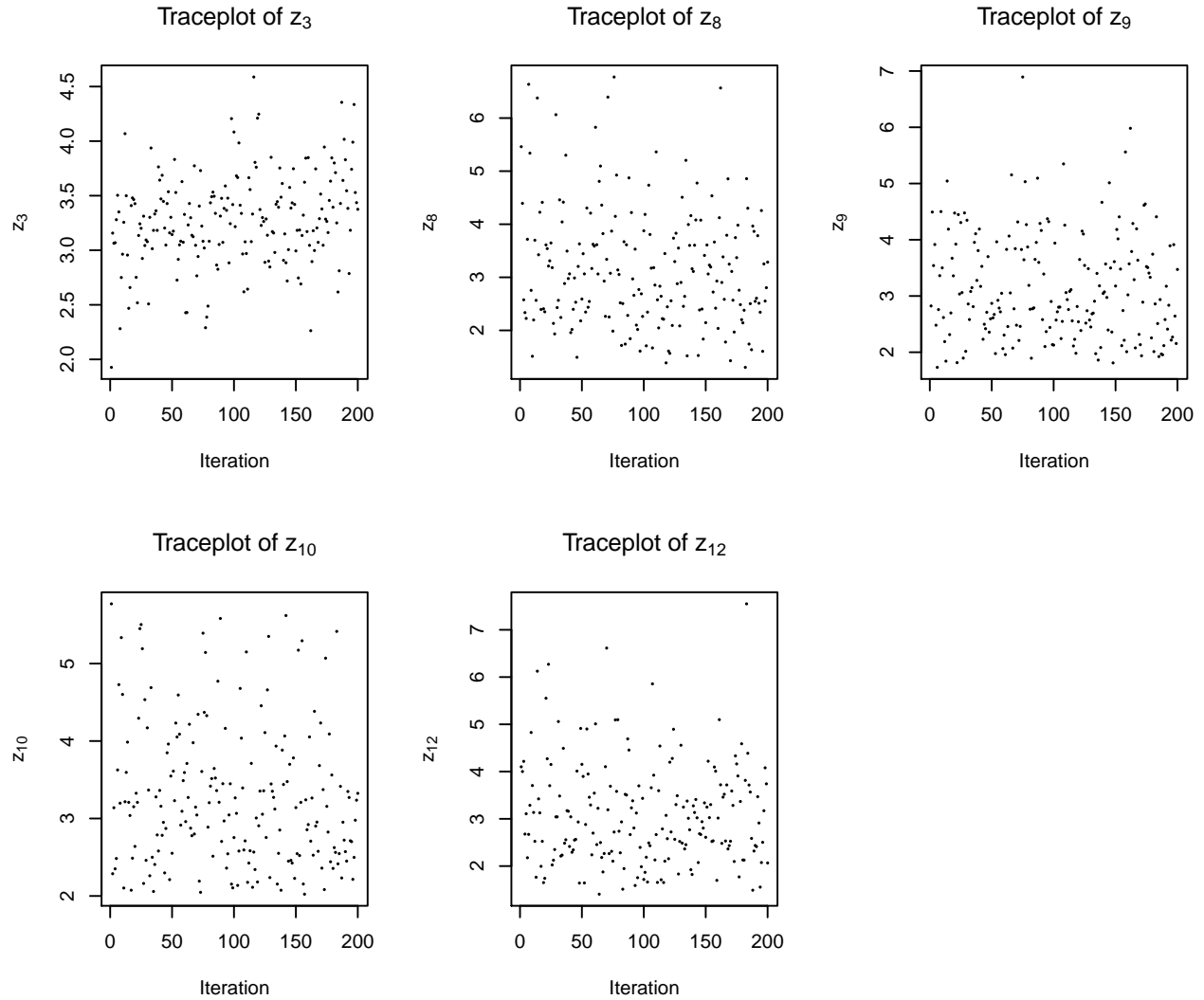


Figure 2: Traceplot of $z_3, z_8, z_9, z_{10}, z_{12}$.

```

#Create Running Average Plot for Theta
plot(1:n.iter, run.avg[,1], type = "l",
xlab = "Iteration", ylab = expression(theta),
main = expression(paste("Running Average Plot of ", theta, " for 200 Iterations")))

#Create Running Average plots for Censored Values
par(mfrow=c(2,3))
missing.index <- c(3,8,9,10,12)
for (ind in missing.index){
  x.lab <- bquote(z[.(ind)])
  plot(1:n.iter, run.avg[,which(missing.index == ind)], type = "l",
xlab = "Iteration", ylab = x.lab,
main = bquote(paste("Running Average Plot of ", .(x.lab))))
}

```

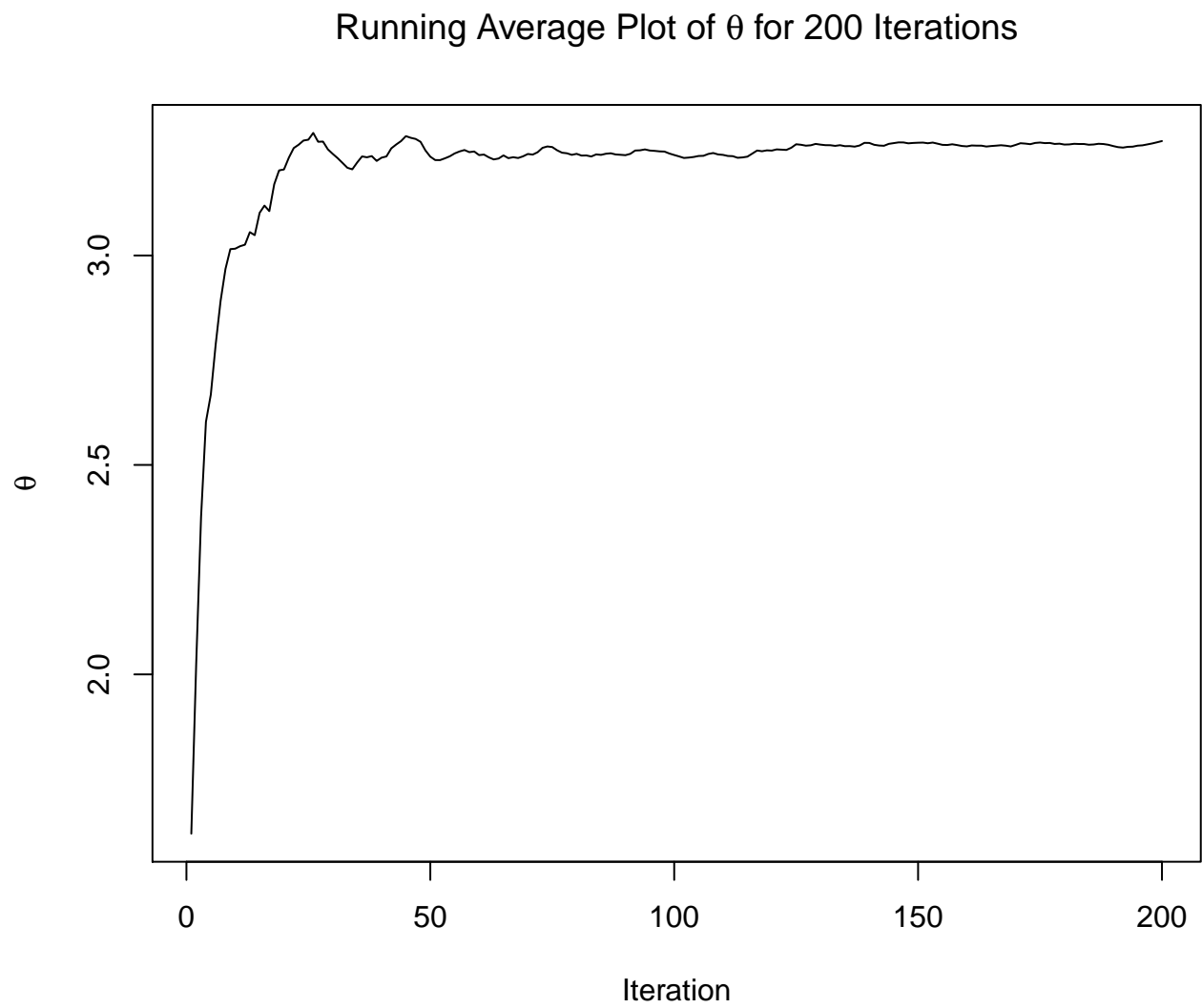


Figure 3: Running average plot of theta

```
}  
plot.new()
```

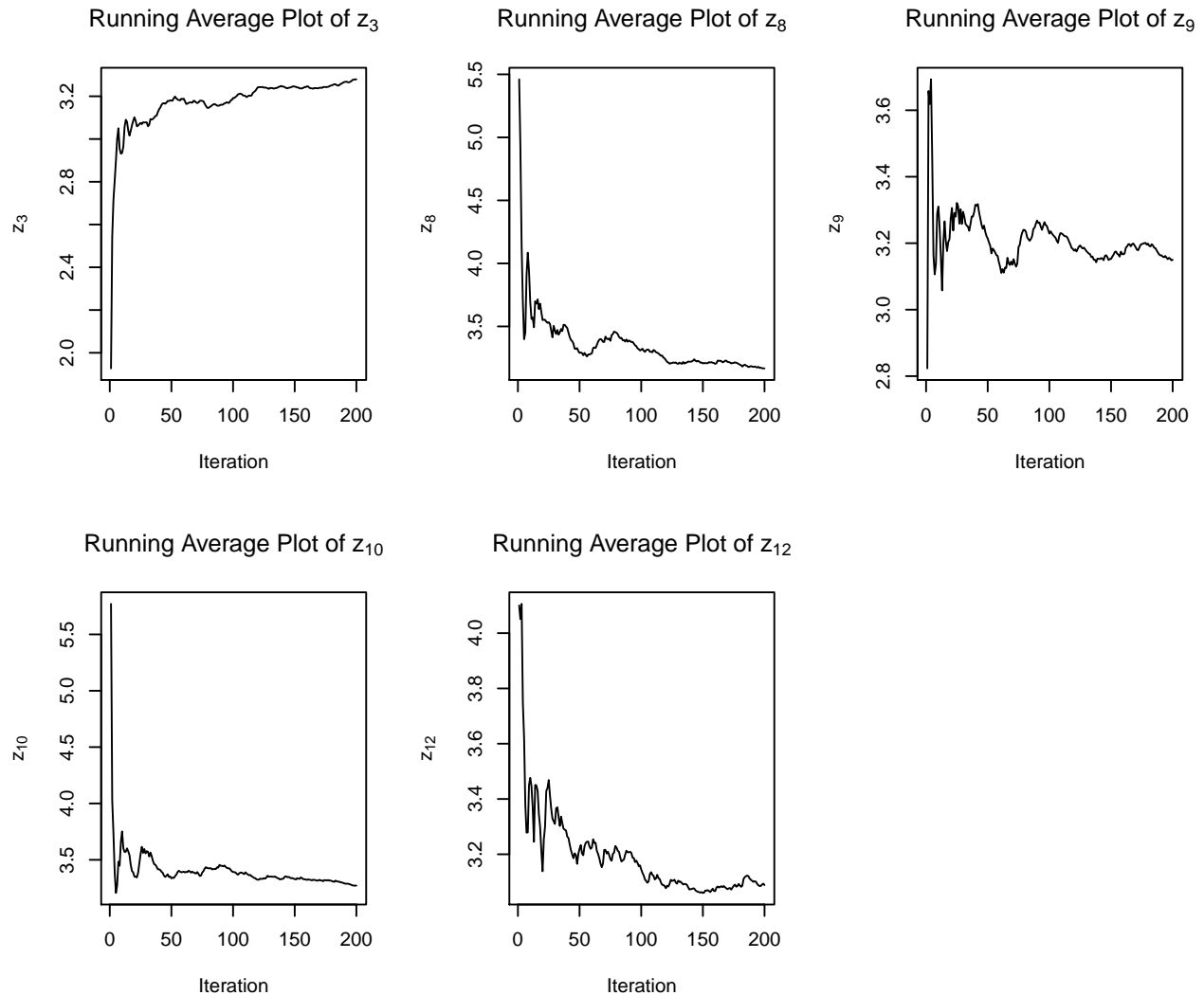


Figure 4: Running average plots of $z_3, z_8, z_9, z_{10}, z_{12}$.

From the plots above, we can see that the plots show clear variation among values which means the sampler is having mixing issues. If you look at the traceplots, there's very visible variation in the values for theta and $z_3, 8, 9, 10, 12$ and the running average plots don't appear to be steady (like a straight line), instead we can see bumps and obvious variation. This means we have not ran our sampler long enough.

Part 2

```
n.iter2 <- 10000  
init.missing2 <- rgamma(length(c), shape = r, rate = init.theta)  
# run sampler  
res2 <- sampleGibbs(z, c, n.iter2, init.theta, init.missing2, r, a, b)  
# get running averages  
run.avg2 <- apply(res2, 2, cumsum)/(1:n.iter2)  
  
#Create Traceplot for Theta  
plot(1:n.iter2, res2[,1], pch = 16, cex = .35,
```

```
xlab = "Iteration", ylab = expression(theta),
main = expression(paste("Traceplot of ", theta, " for 10,000 Iterations")))
```

Traceplot of θ for 10,000 Iterations

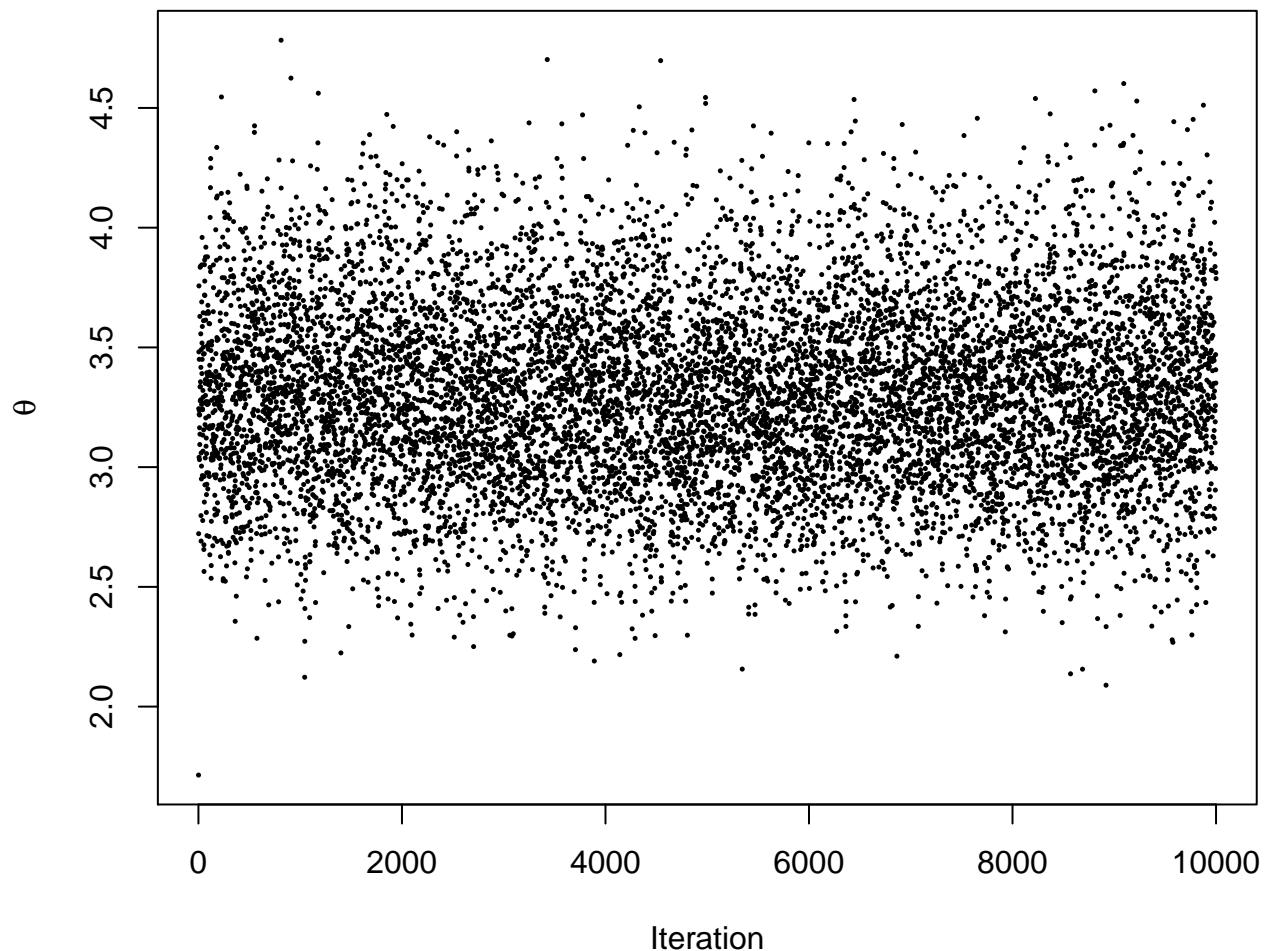


Figure 5: Traceplot of theta

```
#Create Traceplots for Censored Values
missing.index <- c(3,8,9,10,12)
par(mfrow=c(2,3))
for (ind in missing.index){
  x.lab <- bquote(z[.(ind)])
  plot(1:n.iter2, res2[,which(missing.index == ind)], pch = 16, cex = .35,
       xlab = "Iteration", ylab = x.lab,
       main = bquote(paste("Traceplot of ", .(x.lab))))
}
plot.new()

#Create Running Average Plot for Theta
plot(1:n.iter2, run.avg2[,1], type = "l",
     xlab = "Iteration", ylab = expression(theta),
     main = expression(paste("Running Average Plot of ", theta, " for 10,000 iterations")))
```

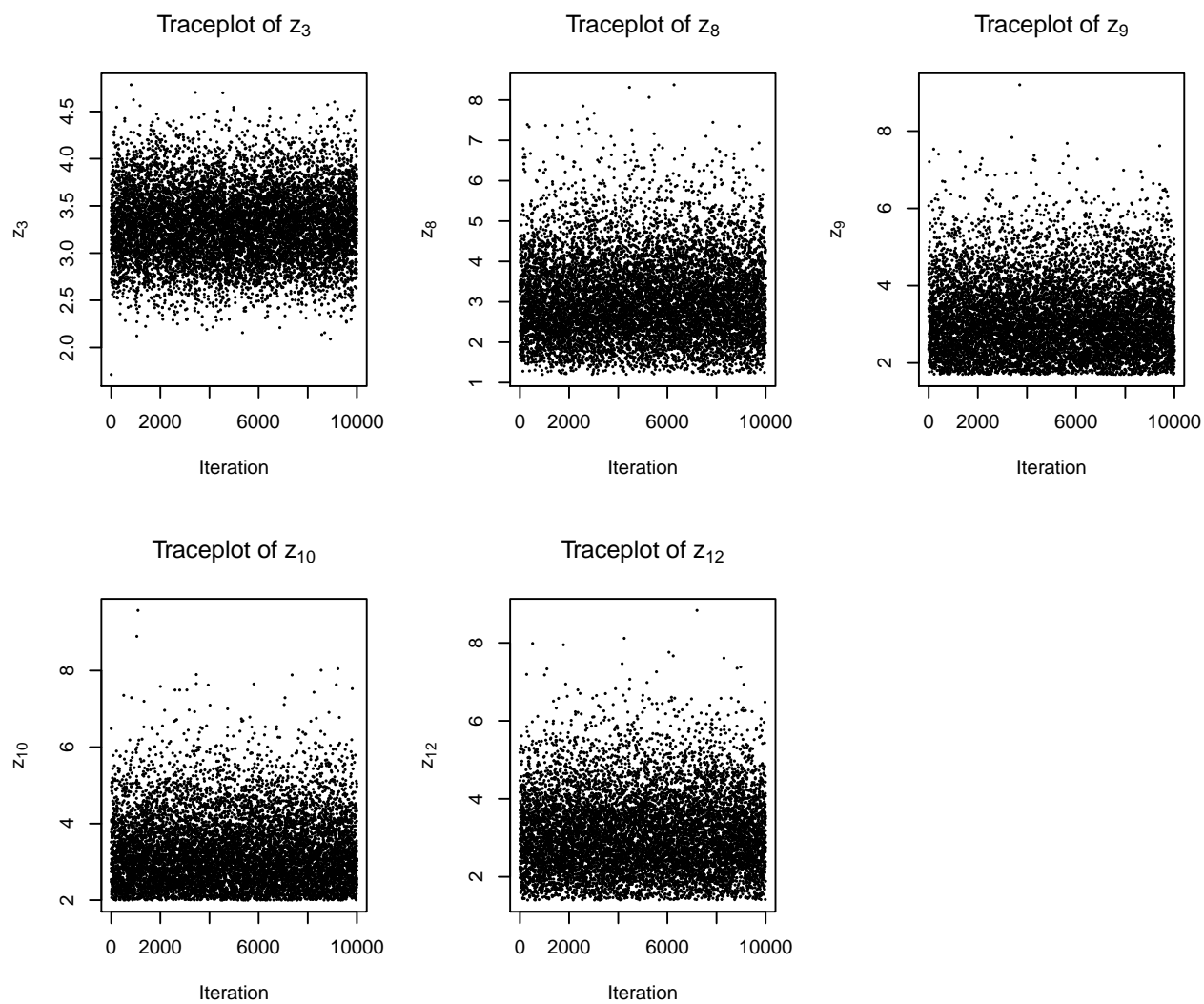


Figure 6: Traceplot of $z_3, z_8, z_9, z_{10}, z_{12}$.

Running Average Plot of θ for 10,000 iterations

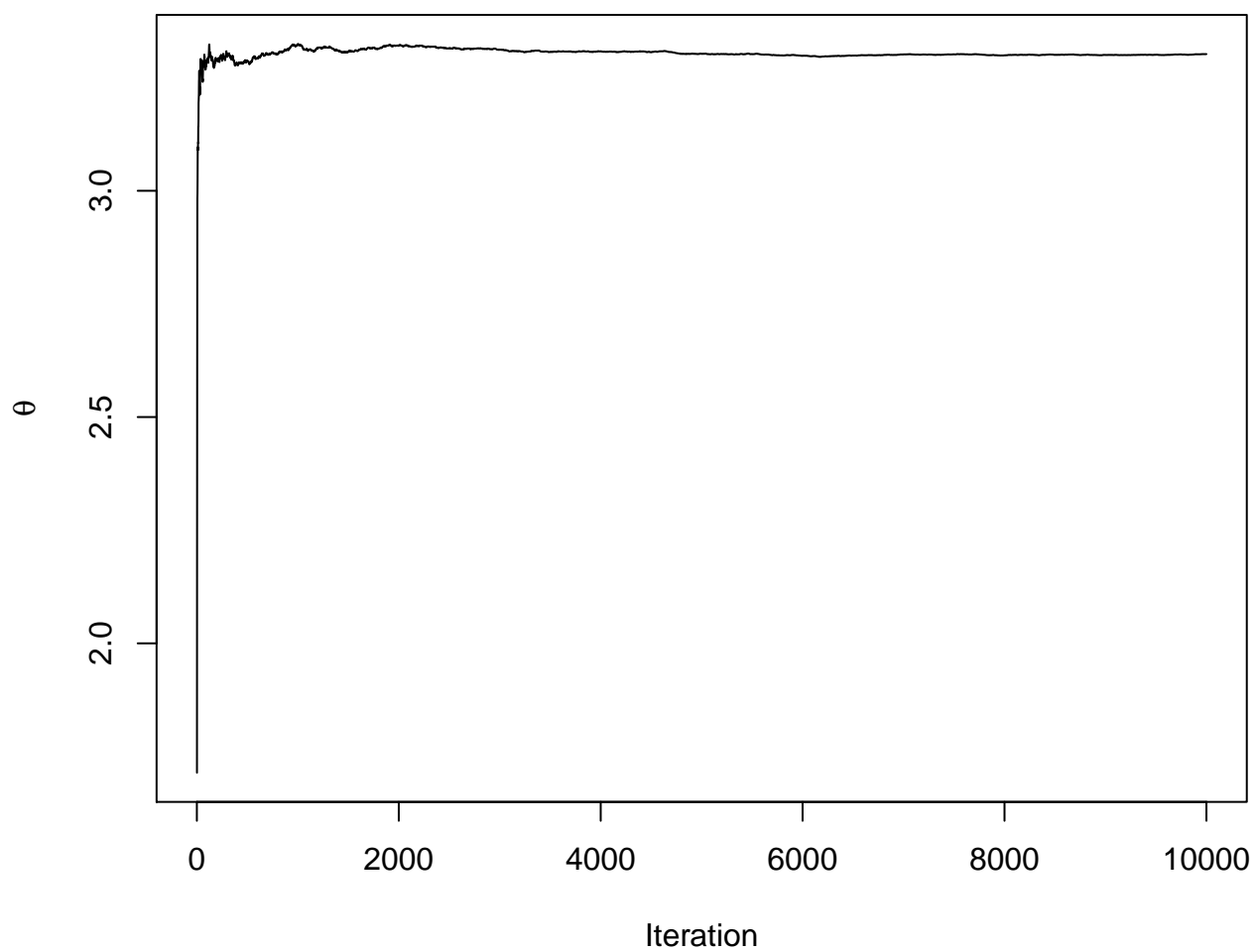


Figure 7: Running average plot of theta


```

#Create Running Average plots for Censored Values
par(mfrow=c(2,3))
missing.index <- c(3,8,9,10,12)
for (ind in missing.index){
  x.lab <- bquote(z[.(ind)])
  plot(1:n.iter2, run.avg2[,which(missing.index == ind)], type = "l",
       xlab = "Iteration", ylab = x.lab,
       main = bquote(paste("Running Average Plot of ", .(x.lab))))
}
plot.new()

```

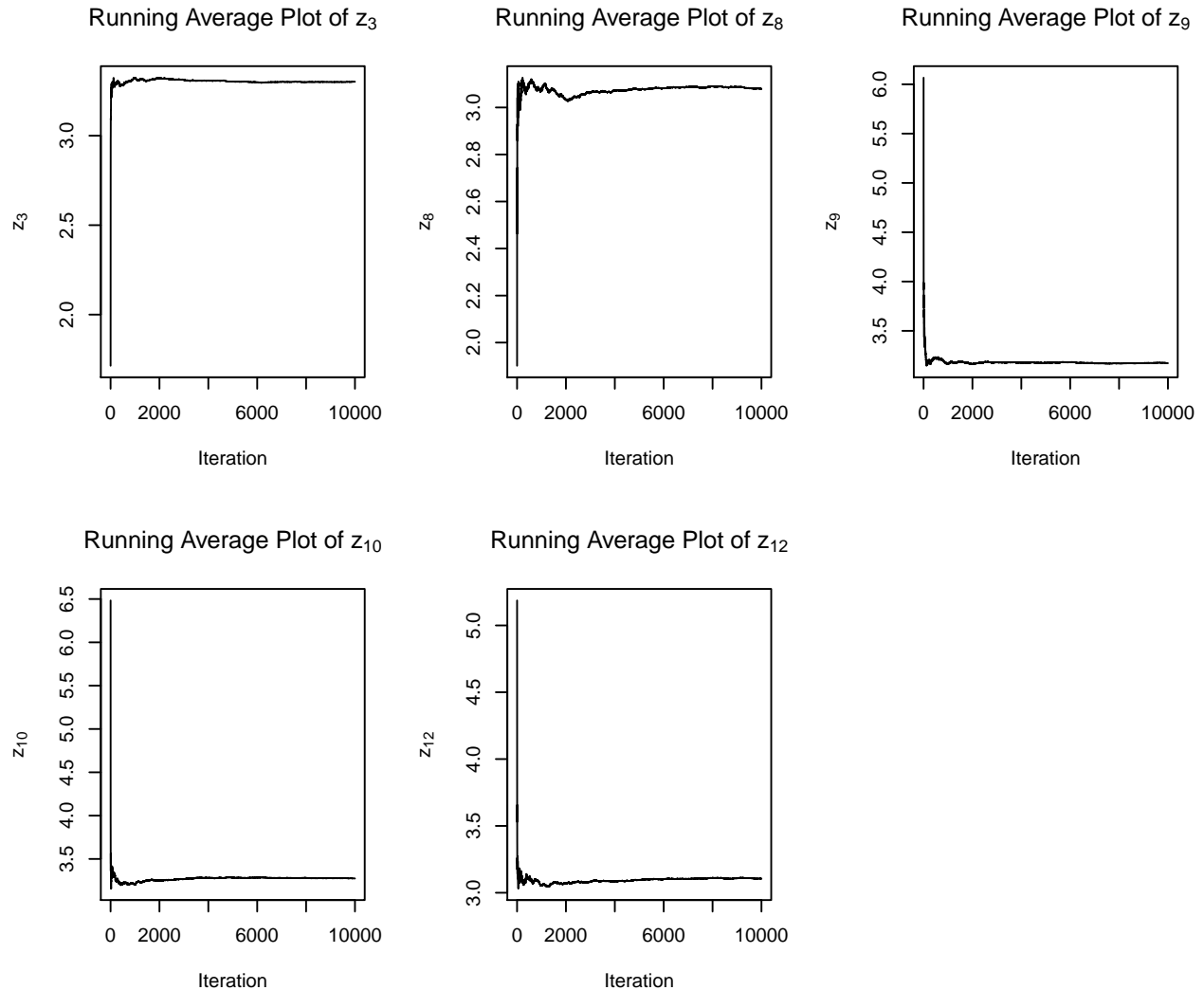


Figure 8: Running average plots of $z_3, z_8, z_9, z_{10}, z_{12}$.

From the plots above, we see a significant improvement from the 200 iterations plots. We can see that in the traceplot for theta, that the majority of the datapoints take on values between 3.0-3.5. The plot looks like it's centered around 3.25 approximately. The same can be said about the traceplots for z_3, z_8, z_9, z_{10} , and z_{12} where you can now see a heavy concentration of points around certain values with the majority of z_3 being between 3.0-3.5, z_8 being between 2-3.5, z_9, z_{10} and z_{12} being between 2-4. The running average plots look far more steady (like a straight line). With these plots, it seems like we have run our sampler enough but we still don't know whether the chain has converged for sure.

Part 3

```
# density plots
plot(density(res2[,1]), xlab = expression(theta),
     main = expression(paste("Density of ", theta)))
abline(v = mean(res2[,1]), col = "red")
```

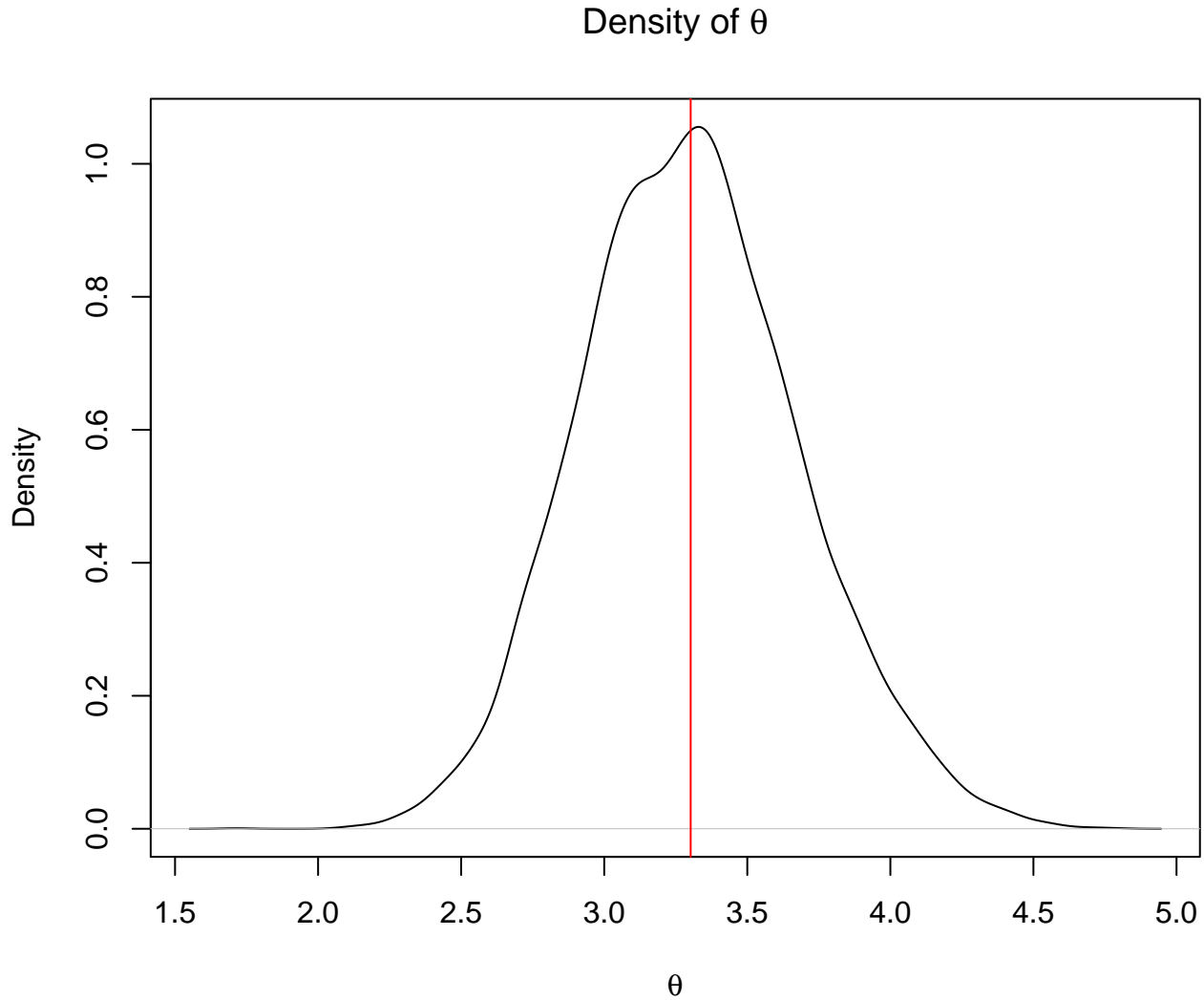


Figure 9: Estimated posterior density of theta

```
plot(density(res2[,4]), xlab = expression(z[9]),
     main = expression(paste("Density of ", z[9])))
abline(v = mean(res2[,4]), col = "red")
```

From the density plot of theta, we can see that the rate parameter for the life time distribution is around 3.25. This means the average life expectancy (r/θ with r being 10) is around 3 years ($10/3.25$) and the variance (r/θ^2) being around 1 ($10/10.5625$). For the density plot of z_9 , we can see the the value of z_9 with the highest density is around 3. This means, we predict that patient 9 who lived at least 2 years after the operation, most likely, to have made it to 3 years. Because we don't know what actually happened to patient 2, we use our latent variable z_9 to impute a value instead. And the density plot of z_9 reveals that a value around 3 is far more likely than values such as 6 or 8 years.

Part 4

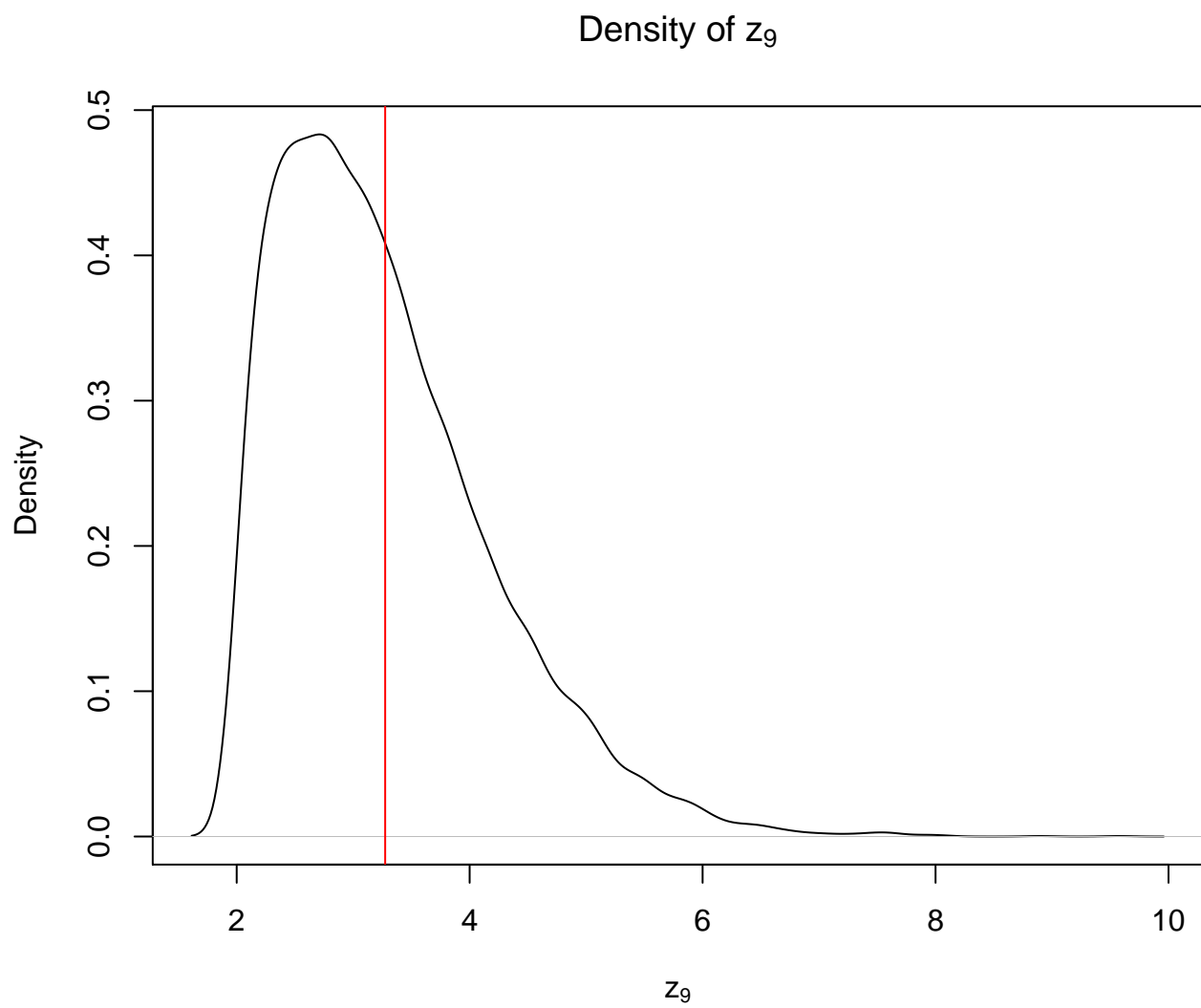


Figure 10: Estimated posterior density of z_9 (posterior mean in red).

```

b2 <- 100
# run sampler
res3 <- sampleGibbs(z, c, n.iter2, init.theta, init.missing, r, a, b2)

# density plots
plot(density(res3[,1]), xlab = expression(theta),
     main = expression(paste("Density of ", theta, " with b = 100")))
abline(v = mean(res3[,1]), col = "red")

```

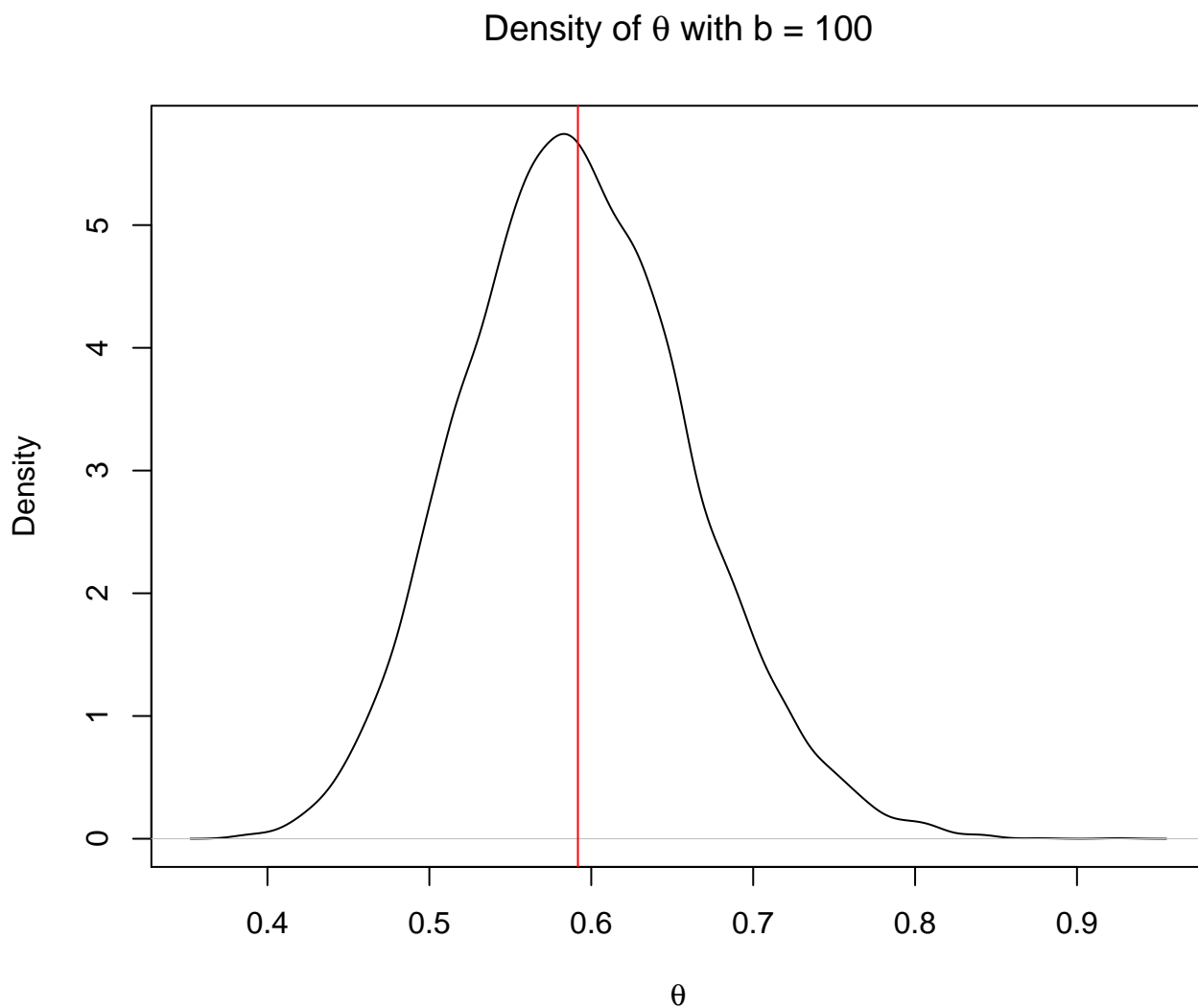


Figure 11: Estimated posterior density of theta

```

plot(density(res3[,4]), xlab = expression(z[9]),
     main = expression(paste("Density of ", z[9], " with b = 100")))
abline(v = mean(res3[,4]), col = "red")

a2 <- 100
# run sampler
res4 <- sampleGibbs(z, c, n.iter2, init.theta, init.missing, r, a2, b)

# density plots
plot(density(res4[,1]), xlab = expression(theta),

```

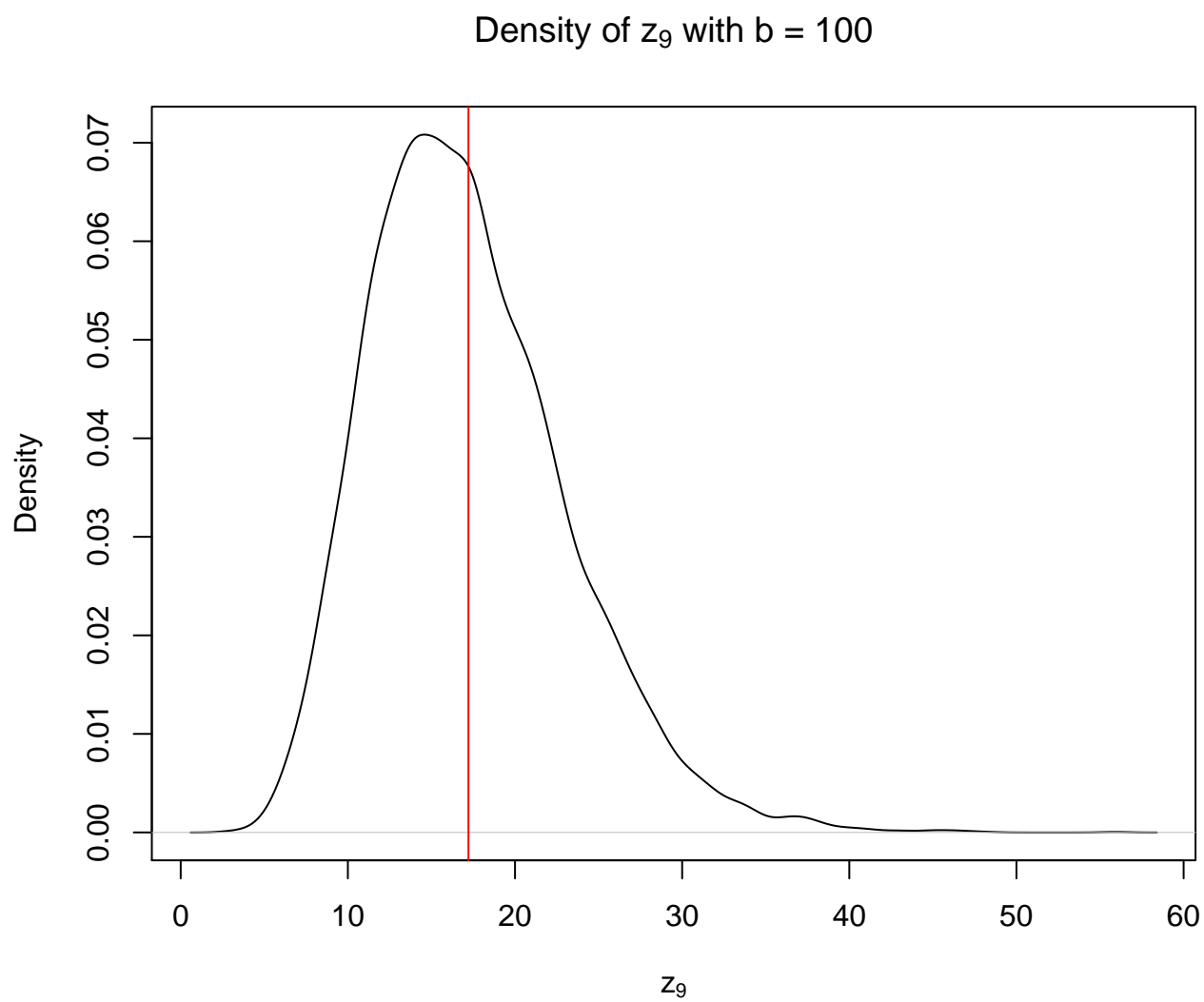


Figure 12: Estimated posterior density of z_9 (posterior mean in red).

```
main = expression(paste("Density of ", theta, " with a = 100"))
abline(v = mean(res4[,1]), col = "red")
```

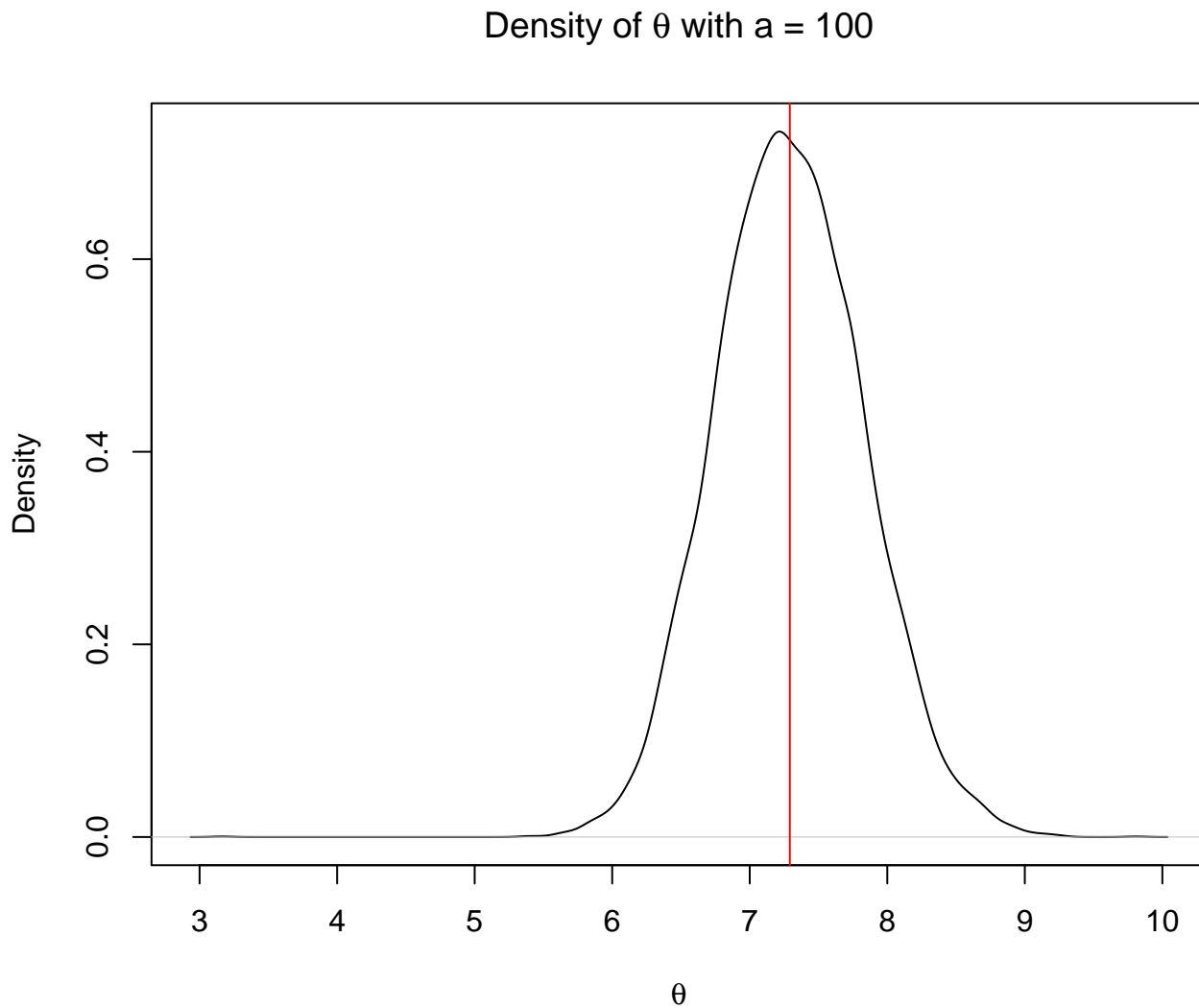
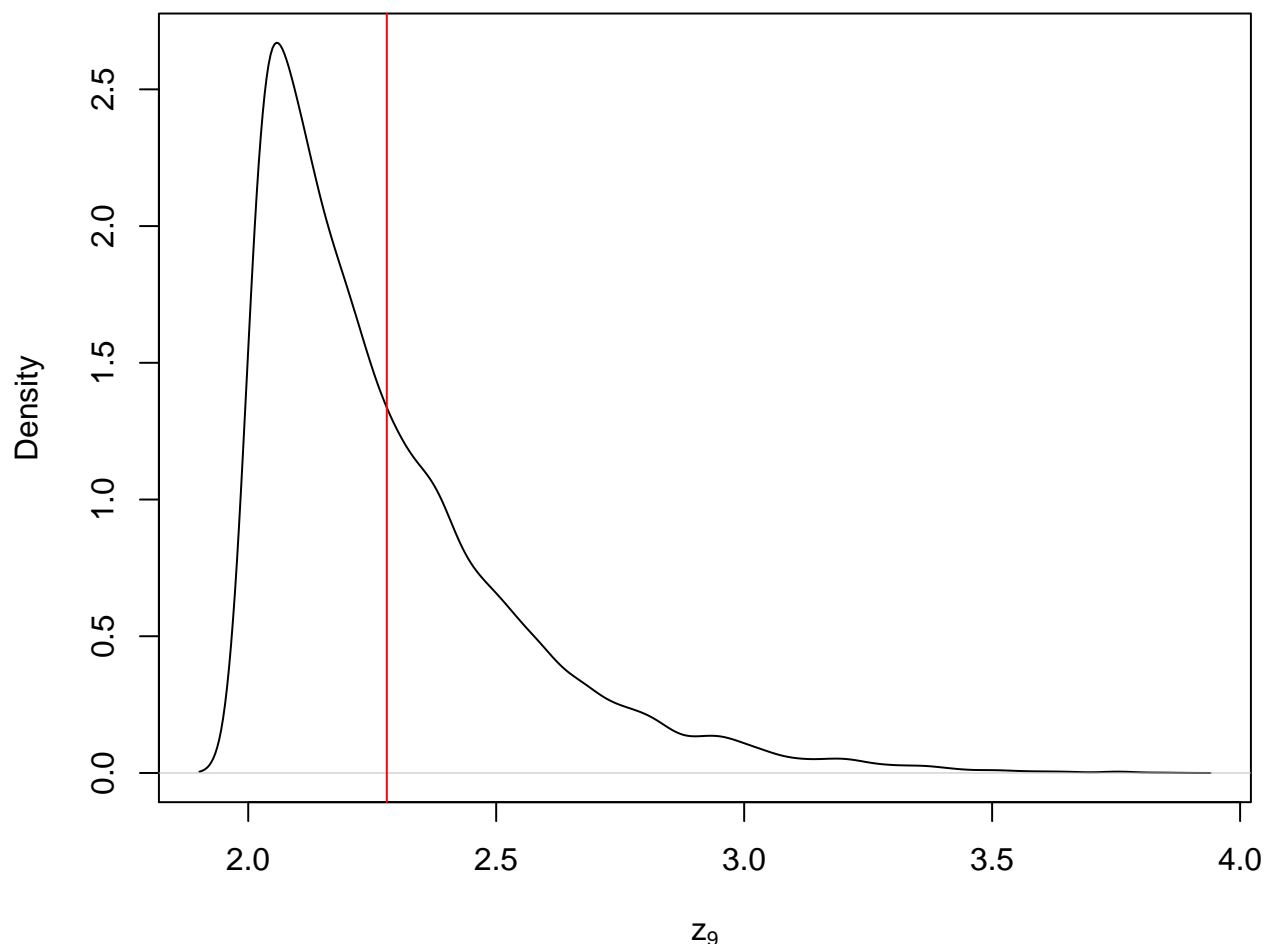


Figure 13: Estimated posterior density of theta

```
plot(density(res4[,4]), xlab = expression(z[9]),
     main = expression(paste("Density of ", z[9], " with a = 100")))
abline(v = mean(res4[,4]), col = "red")
```

Density of z_9 with $a = 100$



When b (or the rate of our prior) is increased, we can see that the mean value of θ (the rate used in our truncated gamma) decreases to around a value of 0.6. When the rate decreases for the truncated gamma, we can see that the mean value of z_9 increases to around 17. When you increase the value of a or the shape of our prior on θ , we see that the mean value of θ increases to around 7. When the rate increases for the truncated gamma, we see the mean value for z_9 decrease to around 2.2. By changing a, b which in turn changes the value of θ , which is used to produce our truncated gamma we sample from for our censored values, our imputed values are affected.