

# STAT 4224 HW #7

Carlyle Morgan

12/20/2021

```
set.seed(2131)
knitr::opts_chunk$set(echo = TRUE)
library(mcmcse)
```

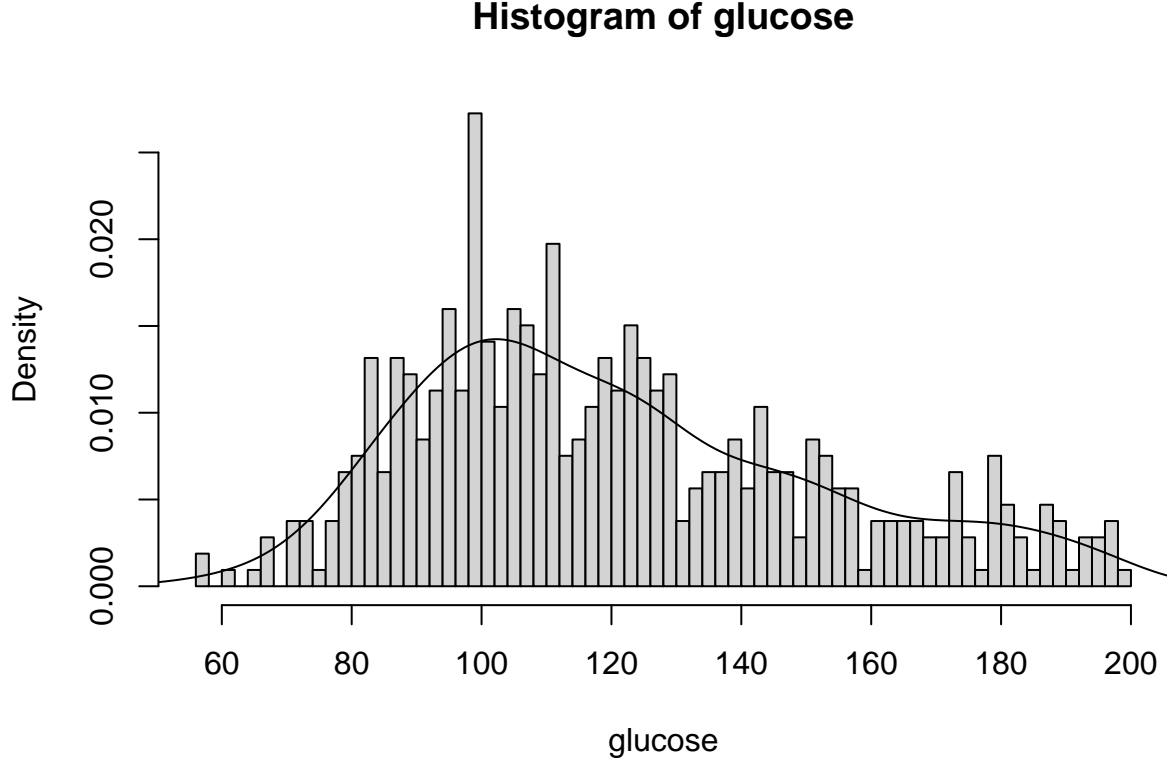
```
## Warning: package 'mcmcse' was built under R version 4.0.5
```

```
library(MASS)
```

1.

a.

```
glucose = scan("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/glucose.dat")
hist(glucose, probability = T, breaks = 100)
lines(density(glucose))
```



The data seems to be right skewed. Meanwhile, the normal distribution is symmetric.

**b.**

For  $j = 1, 2$  and  $\kappa_0 = 1$ ,

$$\sigma_j^2 | y, z \sim Inv - \chi^2(\nu_0 + n_j, \frac{1}{\nu_0 + n_j} [\nu_0 \sigma_0^2 + \sum_{i=1}^2 (y_i - \bar{y}_j)^2 + \frac{\kappa_0 n_j}{\kappa_0 + n_j} (\bar{y}_j - \mu_0)^2])$$

and

$$\mu_j | y, z, \sigma_j^2 \sim N(\frac{\kappa_0 \mu_0 + n_j \bar{y}_j}{\kappa_0 + n_j}, \frac{\sigma_j^2}{\kappa_0 + n_j})$$

Since  $\lambda$  only interacts directly with  $z_i$ ,  $p(\lambda | y, z, \mu, \sigma) = p(\lambda | z)$ . This is a simple beta-binomial model:  $\lambda | z \sim Beta(a + n_1, b + n_2)$ .

Lastly,  $p(z | y, \lambda, \mu, \sigma) \propto p(z | \lambda, \mu, \sigma) p(y | z, \lambda, \mu, \sigma) = p(z | \lambda) p(y | z, \mu, \sigma)$ .

$p(z_i | \lambda)$  is defined by the problem, and  $p(y | z, \mu, \sigma)$  is a likelihood.

Let  $Z_{1i} := \lambda dnorm(y_i; \mu_1, \sigma_1^2)$

and  $Z_{2i} := (1 - \lambda) dnorm(y_i; \mu_1, \sigma_1^2)$

$$\therefore p(z_i = 1 | y, \lambda, \mu, \sigma) = \frac{Z_{i1}}{Z_{i1} + Z_{i2}}$$

and  $z_i - 1 | y, \lambda, \mu, \sigma \sim Ber(\frac{Z_{i2}}{Z_{i1} + Z_{i2}})$

c.

```

a<-1;b<-1;mu.0<-120;nu.0<-1;kappa.0<-1;sigma2.0<-1000;lambda.0<-rbeta(1,1,1)
z.0<-rbinom(length(glucose),1,1-lambda.0)+1; S<-10000; y<-glucose

nj<-function(z,j){length(z[z==j])}
ybarj<-function(y,z,j){mean(y[z==j])}
varj<-function(y,z,j){var(y[z==j])}

musigupdate<-function(y,z,mu.0,nu.0,kappa.0,sigma2.0,j){
  n.j<-nj(z,j); ybar.j<-ybarj(y,z,j); s2.j<-varj(y,z,j);
  nu.n<-nu.0+n.j; kappa.n<-kappa.0+n.j
  sigma2.n<-(1/nu.n)*(nu.0*sigma2.0+(n.j-1)*s2.j+kappa.0*n.j * (ybar.j-mu.0)^2/kappa.n)
  mu.n = (kappa.0*mu.0+n.j*ybar.j)/kappa.n
  sigma2.sim<-1/rgamma(1,nu.n/2,nu.n*sigma2.n/2)
  mu.sim=rnorm(1,mu.n,sqrt(sigma2.sim/kappa.n))
  c(mu.sim,sigma2.sim)
}
lambdaupdate<-function(a,b,z){
  rbeta(1,a+nj(z,1),b+nj(z,2))
}
zupdate<-function(lambda, y,z,mu1,mu2,sigma21,sigma22){
  Z1<-lambda*dnorm(y,mu1,sqrt(sigma21))
  Z2<-(1-lambda)*dnorm(y,mu2,sqrt(sigma22))
  p<-Z2/(Z1+Z2);
  rbinom(length(y),1,p)+1
}
buildchain<-function(S,y,a,b,nu.0,kappa.0,mu.0,sigma2.0,lambda.0,z.0){
  z.chain<-matrix(NA,S,length(y))
  lambda.chain<-rep(0,S)
  mu1.chain<-rep(0,S)
  mu2.chain<-rep(0,S)
  sigma21.chain<-rep(0,S)
  sigma22.chain<-rep(0,S)
  z<-z.0;lambda<-lambda.0;mu1<-mu.0;mu2<-mu.0;sigma21<-sigma2.0;sigma22<-sigma2.0
  for(t in 1:S){
    musig1<-musigupdate(y,z,mu.0,nu.0,kappa.0,sigma2.0,1)
    musig2<-musigupdate(y,z,mu.0,nu.0,kappa.0,sigma2.0,2)
    mu1<-min(musig1[1],musig2[1])
    sigma21<-list(musig1,musig2)[[which.min(c(musig1[1],musig2[1]))]][2]
    mu2<-max(musig1[1],musig2[1])
    sigma22<-list(musig1,musig2)[[which.max(c(musig1[1],musig2[1]))]][2]
    lambda<-lambdaupdate(a,b,z)
    z<-zupdate(lambda,y,z,mu1,mu2,sigma21,sigma22)
    mu1.chain[t]<-mu1;mu2.chain[t]<-mu2;sigma21.chain[t]<-sigma21;
    sigma22.chain[t]<-sigma22;lambda.chain[t]<-lambda;z.chain[t,<-z
  }
  list(mu1.chain=mu1.chain,mu2.chain=mu2.chain,sigma21.chain = sigma21.chain,
       sigma22.chain=sigma22.chain, lambda.chain=lambda.chain, z.chain=z.chain)
}

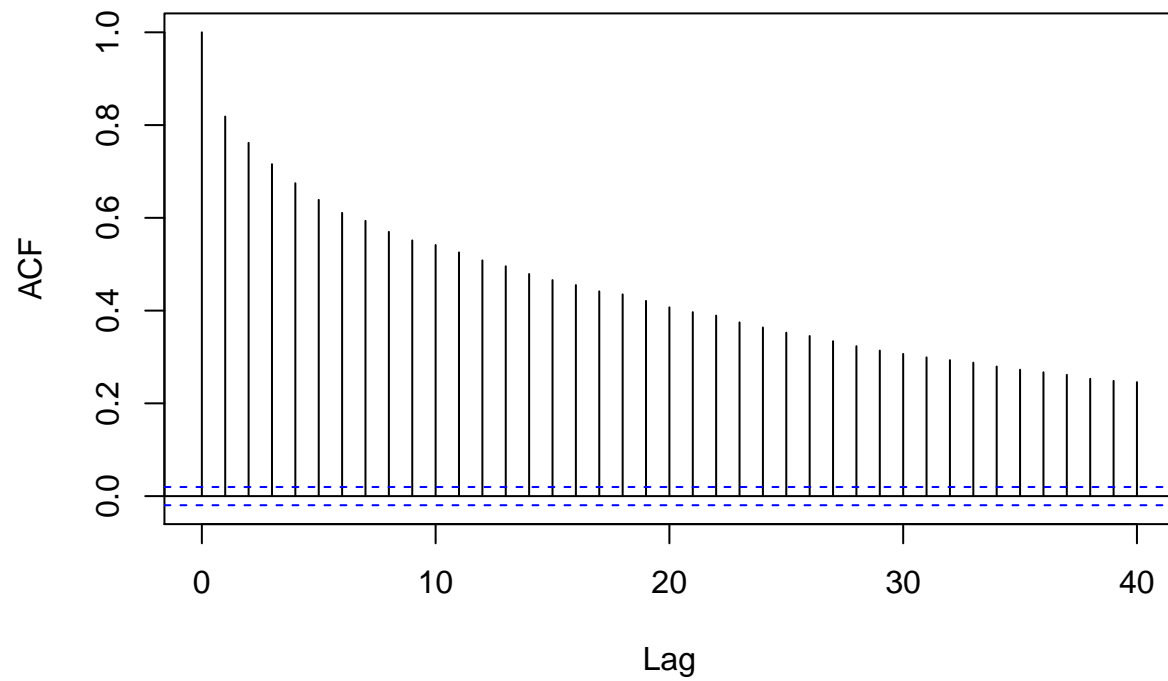
chains<-buildchain(S,y,1,1,nu.0,kappa.0,mu.0,sigma2.0,lambda.0,z.0)
mu1.chain<-chains$mu1.chain
mu2.chain<-chains$mu2.chain

```

```
sigma21.chain<-chains$sigma21.chain  
sigma22.chain<-chains$sigma22.chain  
lambda.chain<-chains$lambda.chain  
z.chain<-chains$z.chain
```

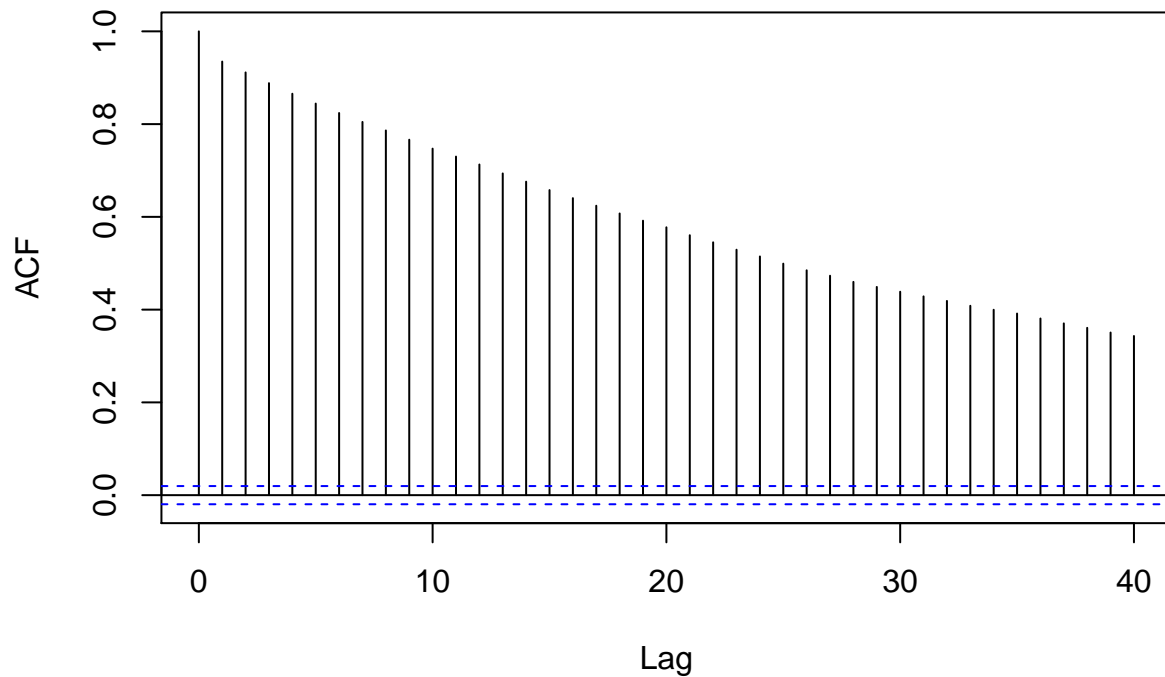
```
acf(mu1.chain)
```

### Series mu1.chain



```
acf(mu2.chain)
```

## Series mu2.chain



```
ess(mu1.chain)
```

```
## [1] 235.547
```

```
ess(mu2.chain)
```

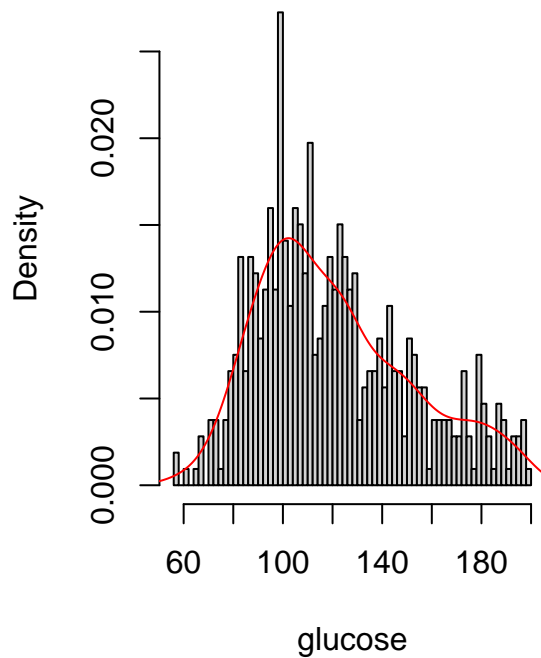
```
## [1] 113.3391
```

d.

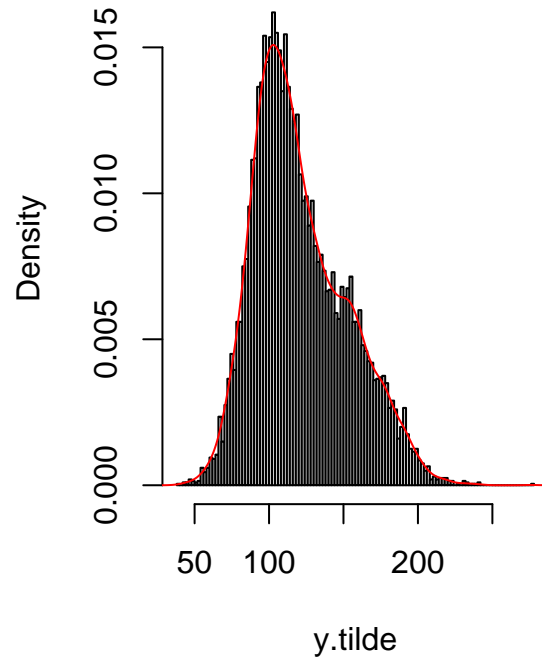
```
z.tilde<-rbinom(length(lambda.chain),1,1-lambda.chain)+1
y.tilde<-rep(NA, length(z.tilde))
for (i in 1:length(y.tilde)){
  if(z.tilde[i]==1){
    y.tilde[i]<-rnorm(1,mu1.chain[i], sqrt(sigma21.chain[i]))
  } else {
    y.tilde[i]<-rnorm(1,mu2.chain[i], sqrt(sigma22.chain[i]))
  }
}
par(mfrow=c(1,2))

hist(glucose, probability = T, breaks = 100)
lines(density(glucose), col = "red")
hist(y.tilde, probability = T, breaks = 100)
lines(density(y.tilde), col = "red")
```

### Histogram of glucose



### Histogram of y.tilde



It looks like the two-component mixture model works pretty well!

## 2.

a.

```
grass<-read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/pdensity.dat", header = T)

density2<-grass$density^2

ids <- sort(unique(grass$plot))
J <- length(ids)
y <- list(); X <- list(); n <- NULL;
for(j in 1:J)
{
y[[j]] <- grass[grass$plot==ids[j], 3]
n[j] <- sum(grass$plot==ids[j])
x.j1 <- grass[grass$plot==ids[j], 2]
x.j1 <- (x.j1 - mean(x.j1))
x.j2 <- density2[grass$plot==ids[j]]
x.j2 <- x.j2-mean(x.j2)
x.j<-cbind(x.j1,x.j2)
X[[j]] <- cbind(rep(1, n[[j]]), x.j)
}
```

```

beta.hat.OLS <- matrix(NA, J, 3)
sigma2.hat <- rep(NA, J)
for(j in 1:J)
{
  fit <- lm(y[[j]] ~ 0 + X[[j]])
  beta.hat.OLS[j,] <- fit$coef
  sigma2.hat <- summary(fit)$sigma^2
  rm(fit)
}

```

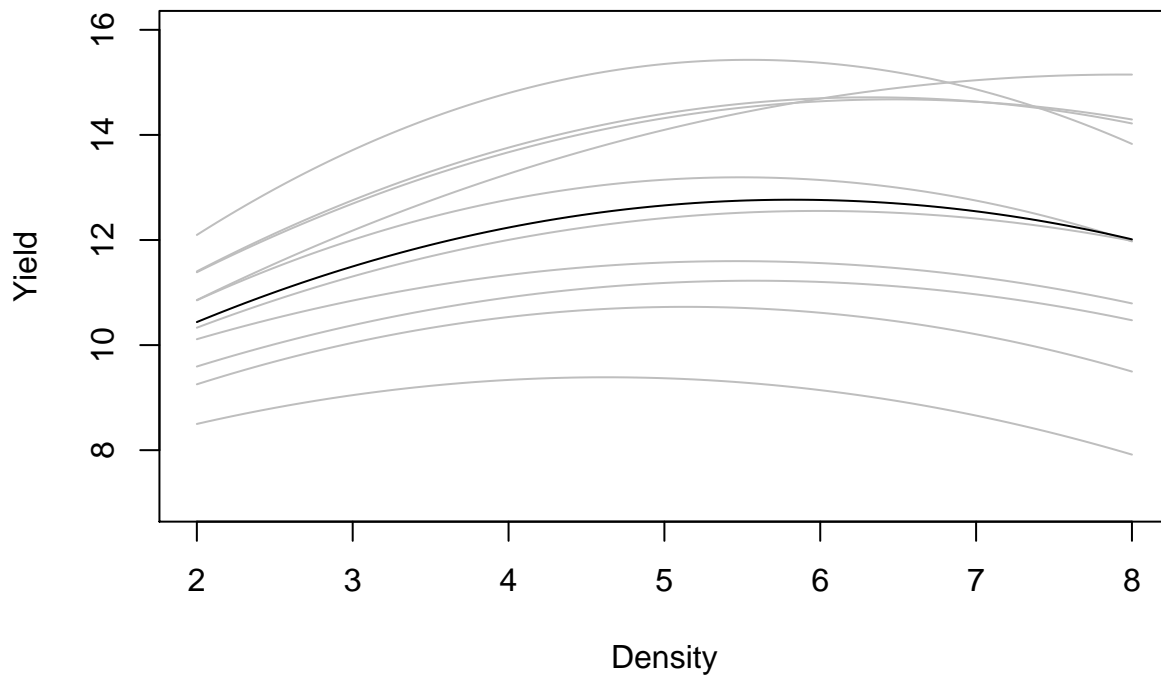
i.

```

dens.plot<-seq(2,8,.01)
X.plot<-cbind(rep(1,length(dens.plot)),dens.plot,dens.plot^2)
plot(c(2,8),range(c(7,16)), type = "n", xlab = "Density", ylab = "Yield", main = "Non-Gibbs OLS")
for (j in 1:J){
  lines(dens.plot, X.plot%*%beta.hat.OLS[j,], col = "gray", type = "l")
}
mean.OLS<-apply(beta.hat.OLS,2,mean)
lines(dens.plot, X.plot%*%mean.OLS, col = "black", type = "l")

```

## Non-Gibbs OLS



ii.

```

mu.hat<-mean.OLS
Sigma.hat<-cov(beta.hat.OLS)

```

```
sigma2.hat<-mean(sigma2.hat)

mu.hat

## [1] 7.36550 1.85485 -0.15925

Sigma.hat

##           [,1]      [,2]      [,3]
## [1,] 0.1653063194 0.06227755 -0.0003503299
## [2,] 0.0622775486 0.27555421 -0.0207426788
## [3,] -0.0003503299 -0.02074268 0.0019684505

sigma2.hat

## [1] 0.8030505
```

b.

```
p<-3
mu.0<-matrix(mu.hat,J,p,byrow = T)
kappa.0<-1
eta.0<-p+2
Lambda.0<-Sigma.hat
nu.0<-1
sigma2.0<-sigma2.hat
T<-10000
beta.chain<-matrix(NA,T,J*p)
sigma2.chain<-rep(NA,T)
mu.chain<-matrix(NA,T,p)
Sigma.chain<-matrix(NA,T,p^2)
beta<-beta.hat.OLS
sigma2<-sigma2.0
mu<-mu.0[1,]
Sigma<-Lambda.0
Sigma.inv<-solve(Sigma)

for(t in 1:T)
{
  for(j in 1:J)
  {
    V.j <- solve( Sigma.inv + 1/sigma2 * t(X[[j]]) %*% X[[j]] )
    m.j <- V.j %*% ( Sigma.inv %*% mu + 1/sigma2 * t(X[[j]]) %*% y[[j]] )
    beta[j,] <- mvrnorm(1, m.j, V.j)
  }
  RSS <- 0
  for(j in 1:J){ RSS <- RSS + sum( (y[[j]] - X[[j]] %*% beta[j,])^2 ) }
  sigma2 <- (nu.0*sigma2.0 + RSS) / rchisq(1, df=nu.0+sum(n))
  beta.bar <- apply(beta, 2, mean)
  beta.bar <- matrix(beta.bar, J, p, byrow=T)
  Sigma.inv <- rWishart(1, eta.0+J, solve(Lambda.0 +
    t(beta - beta.bar) %*% (beta - beta.bar) +
    kappa.0*J / (kappa.0+J) * t(beta.bar - mu.0) %*% (beta.bar - mu.0)))[,,1]
  Sigma <- solve(Sigma.inv); # rm(Sigma.inv);
  mu <- mvrnorm( 1, ( kappa.0 * mu.0[1,] + J * beta.bar[1,] ) / (kappa.0 + J),
```



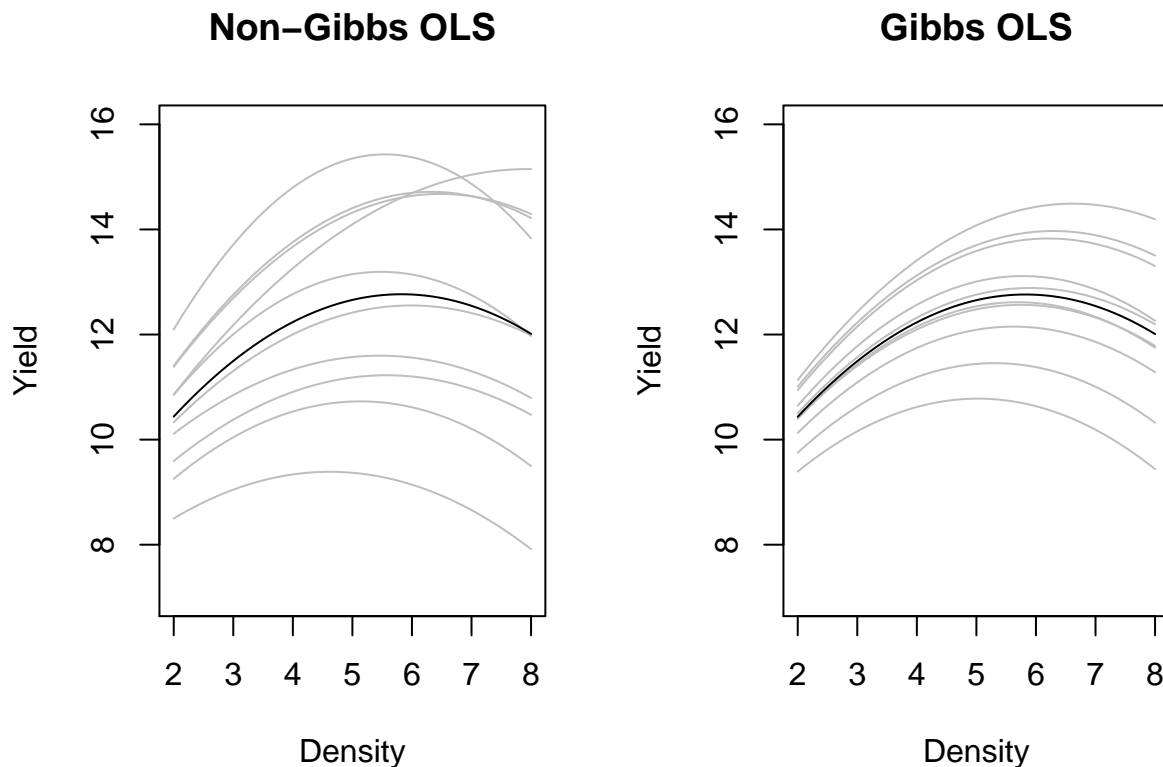
```

1/(kappa.0 + J) * Sigma)
for(j in 1:J){ beta.chain[t, seq(j, j+(p-1)*J, J)] <- beta[j,] }
sigma2.chain[t] <- sigma2;
mu.chain[t,] <- mu;
Sigma.chain[t,] <- c(Sigma);
}
mean.OLS<-apply(beta.hat.OLS,2,mean)
beta.hat<-apply(beta.chain,2,mean)
mu.mean<-apply(mu.chain, 2, mean)

par(mfrow=c(1,2))
plot(c(2,8),range(c(7,16)), type = "n", xlab = "Density", ylab ="Yield", main = "Non-Gibbs OLS")
for (j in 1:J){
  lines(dens.plot, X.plot%%beta.hat.OLS[j,], col = "gray", type = "l")
}

lines(dens.plot, X.plot%%mean.OLS, col = "black", type = "l")
plot(c(2,8),range(c(7,16)), type = "n", xlab = "Density", ylab ="Yield", main = "Gibbs OLS")
for(j in 1:J){
  lines(dens.plot,X.plot%%beta.hat[seq(j,j+(p-1)*J,J)], col = "gray", type = "l")
}
lines(dens.plot,X.plot%%mu.mean,col = "black", type = "l")

```

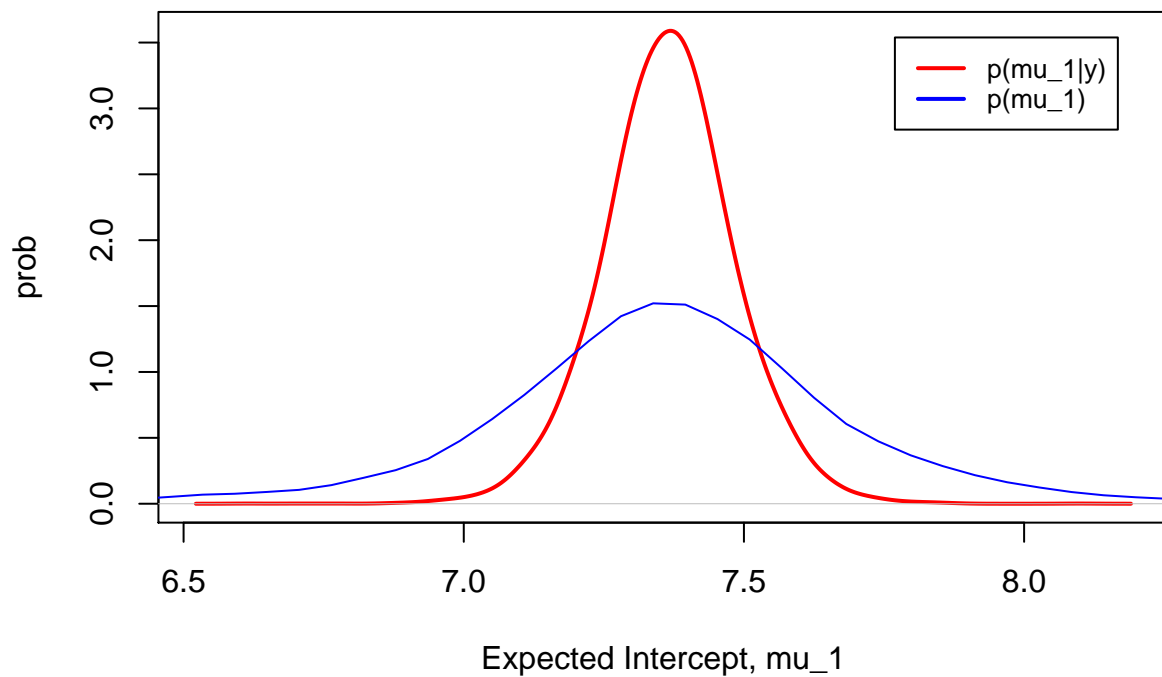


There is less variation in the posterior expectations than in part (a), as the regressions are more correlated due to the hierarchical structure.

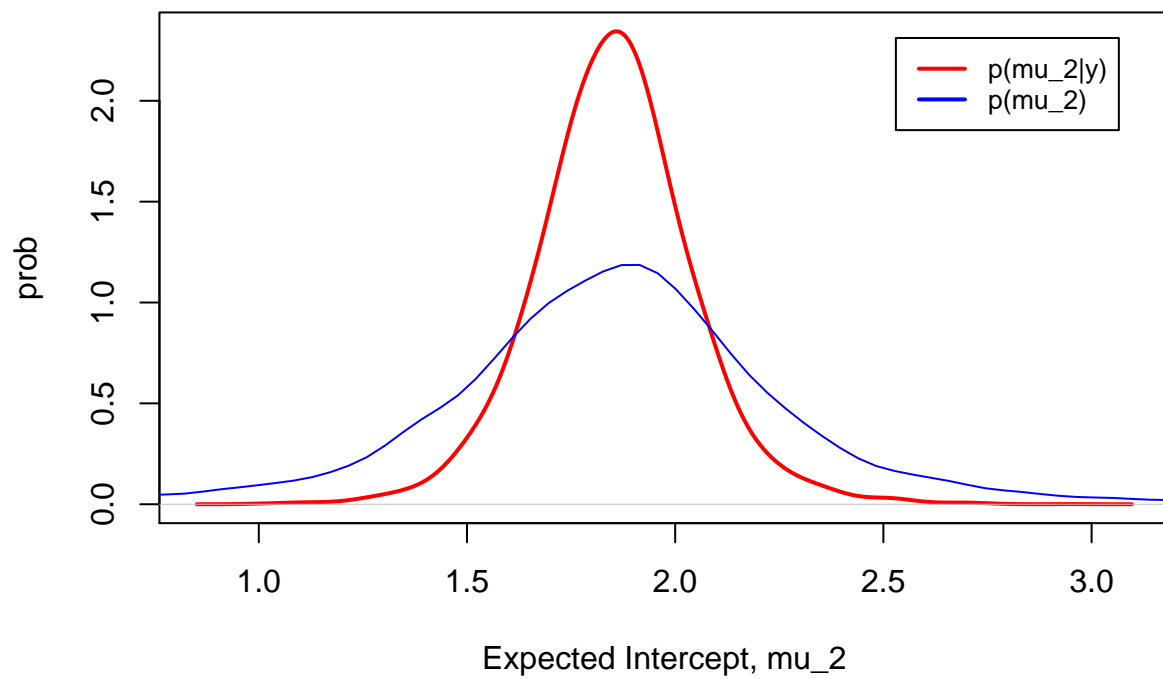
c.

```
mu.prior<-matrix(NA, T, p)
for (t in 1:T){
  Sigma.inv.prior <- rWishart(1, eta.0, solve(Lambda.0))
  Sigma.prior<-solve(matrix(Sigma.inv.prior, nrow = 3))
  mu.prior[t,]<-mvrnorm(1, mu.0[1,],Sigma.prior)
}

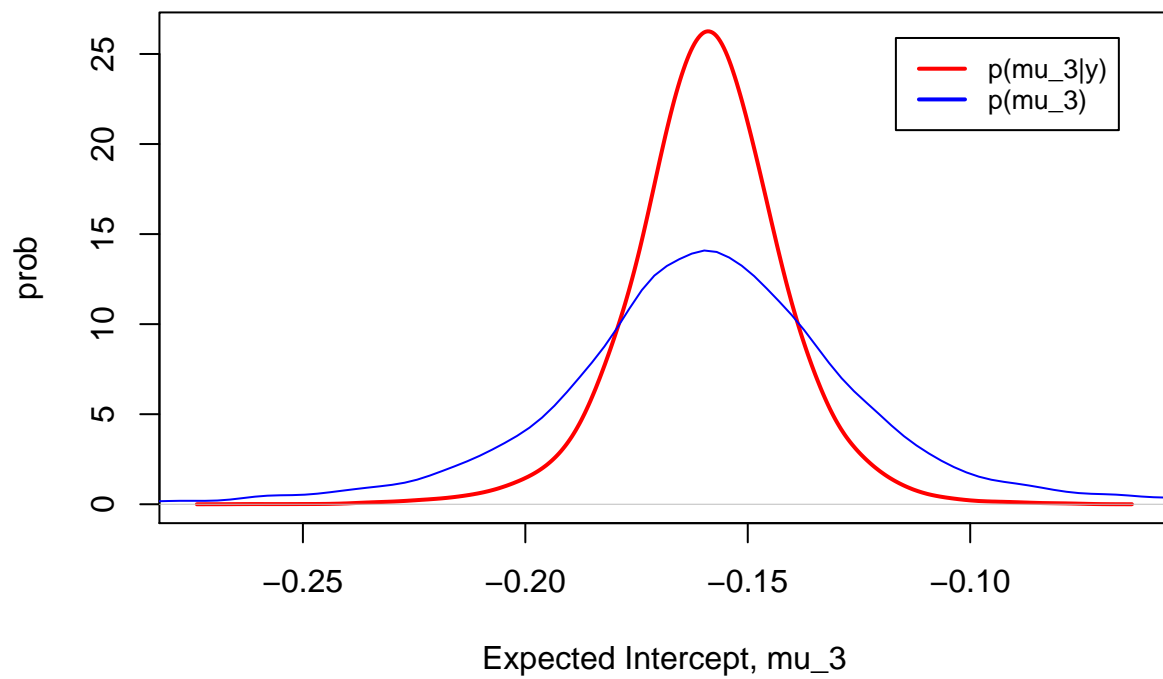
plot(density(mu.chain[,1],adj=2), lwd = 2, main = "", xlab = "Expected Intercept, mu_1", ylab = "prob",
lines(density(mu.prior[,1]), col = "blue")
legend("topright", inset = .05, lty = 1, lwd =2, col = c("red", "blue"), legend = c("p(mu_1|y)", "p(mu_1)"))
```



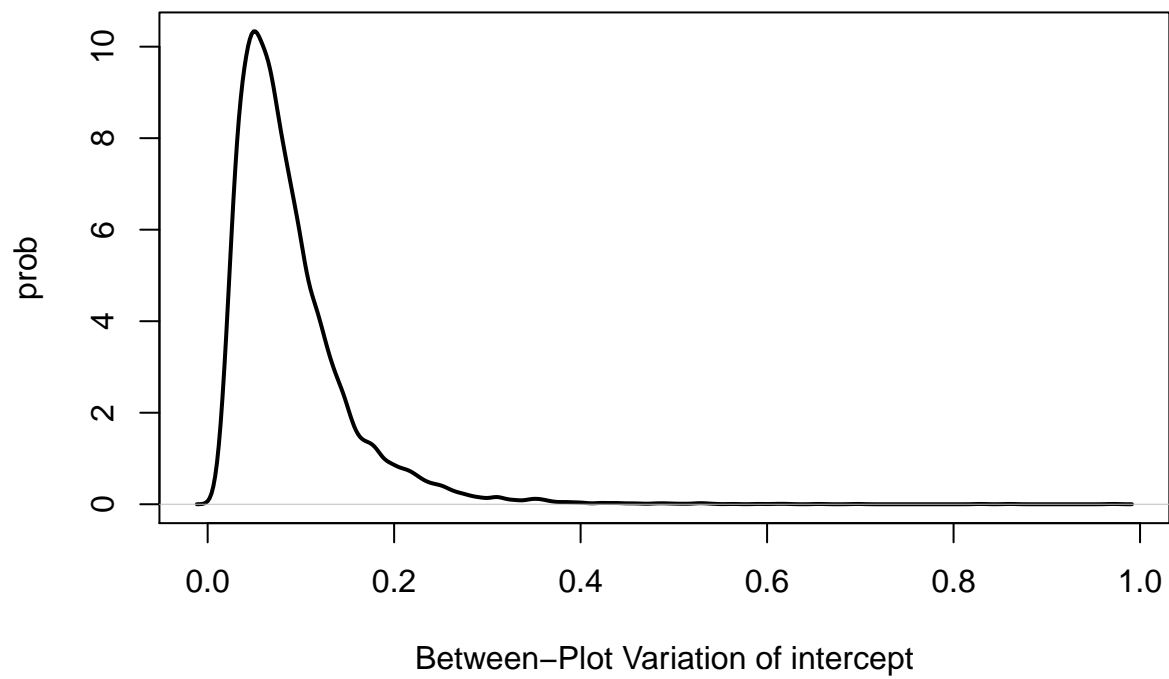
```
plot(density(mu.chain[,2],adj=2), lwd = 2, main = "", xlab = "Expected Intercept, mu_2", ylab = "prob",
lines(density(mu.prior[,2]), col = "blue")
legend("topright", inset = .05, lty = 1, lwd =2, col = c("red", "blue"), legend = c("p(mu_2|y)", "p(mu_2)"))
```



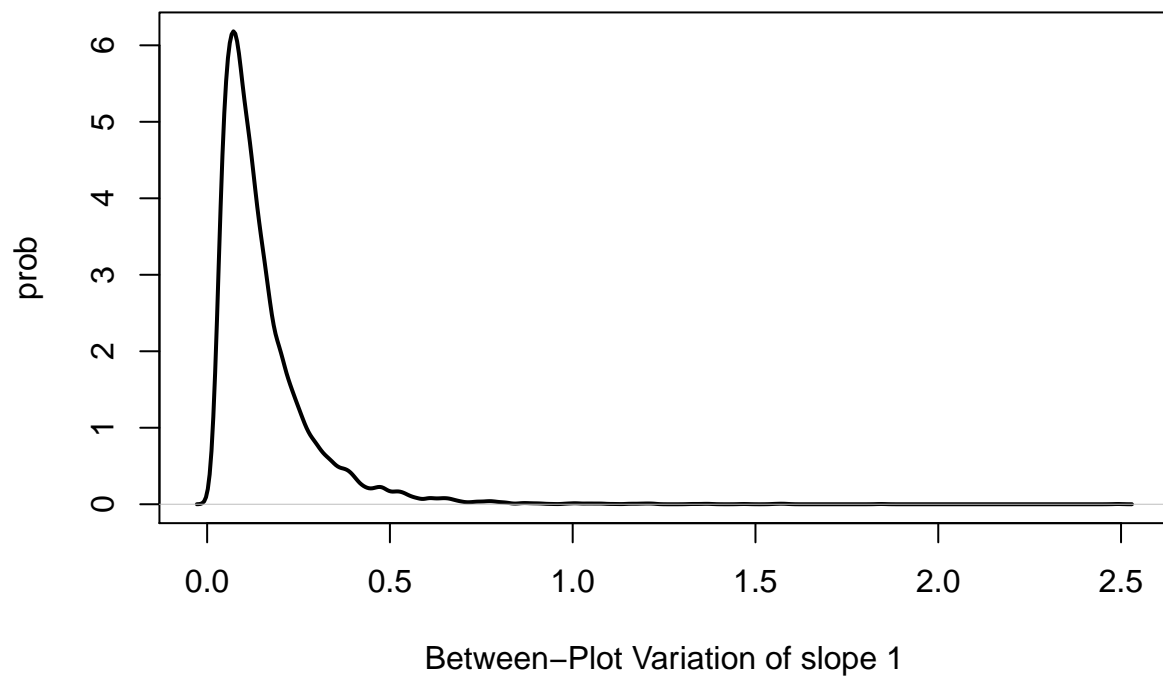
```
plot(density(mu.chain[,3],adj=2), lwd = 2, main = "", xlab = "Expected Intercept, mu_3", ylab = "prob",
lines(density(mu.prior[,3]), col = "blue")
legend("topright", inset = .05, lty = 1, lwd =2, col = c("red", "blue"), legend = c("p(mu_3|y)", "p(mu_3)"))
```



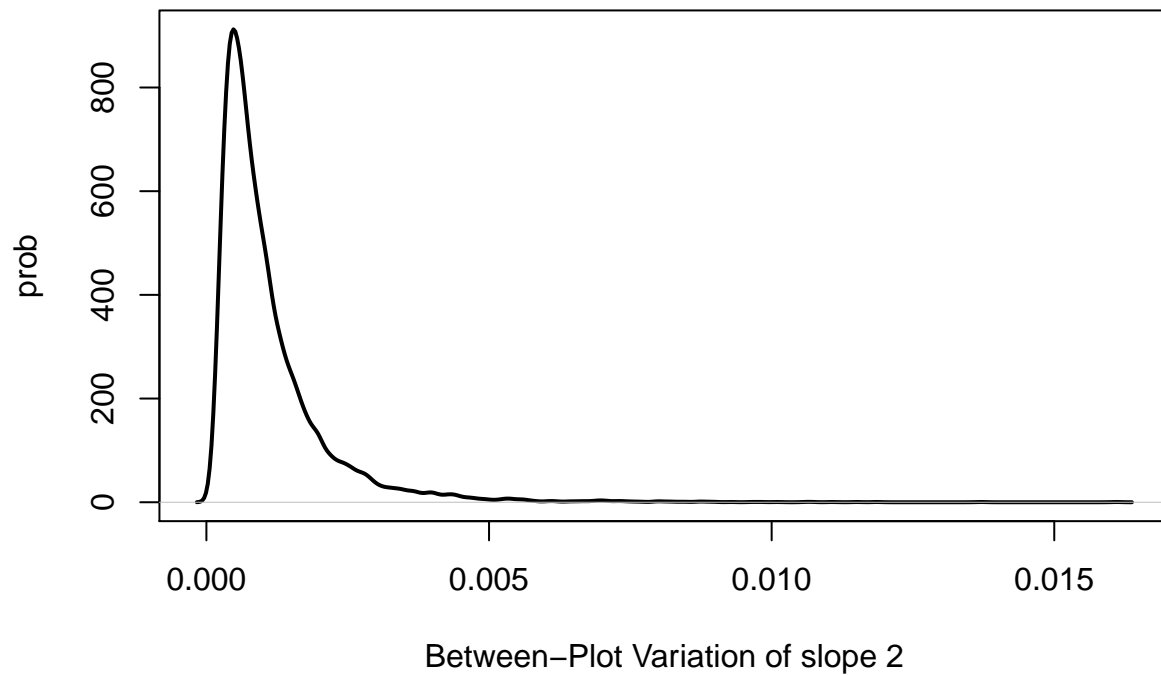
```
diagonals<-Sigma.chain[,c(1,5,9)]
plot(density(diagonals[,1]), lwd = 2, main = "", xlab = "Between-Plot Variation of intercept", ylab = "p(mu_3|y) - p(mu_3)")
```



```
diagonals<-Sigma.chain[,c(1,5,9)]  
plot(density(diagonals[,2]), lwd = 2, main = "", xlab = "Between-Plot Variation of slope 1", ylab = "pr
```



```
diagonals<-Sigma.chain[,c(1,5,9)]
plot(density(diagonals[,3]), lwd = 2, main = "", xlab = "Between-Plot Variation of slope 2", ylab = "pr
```



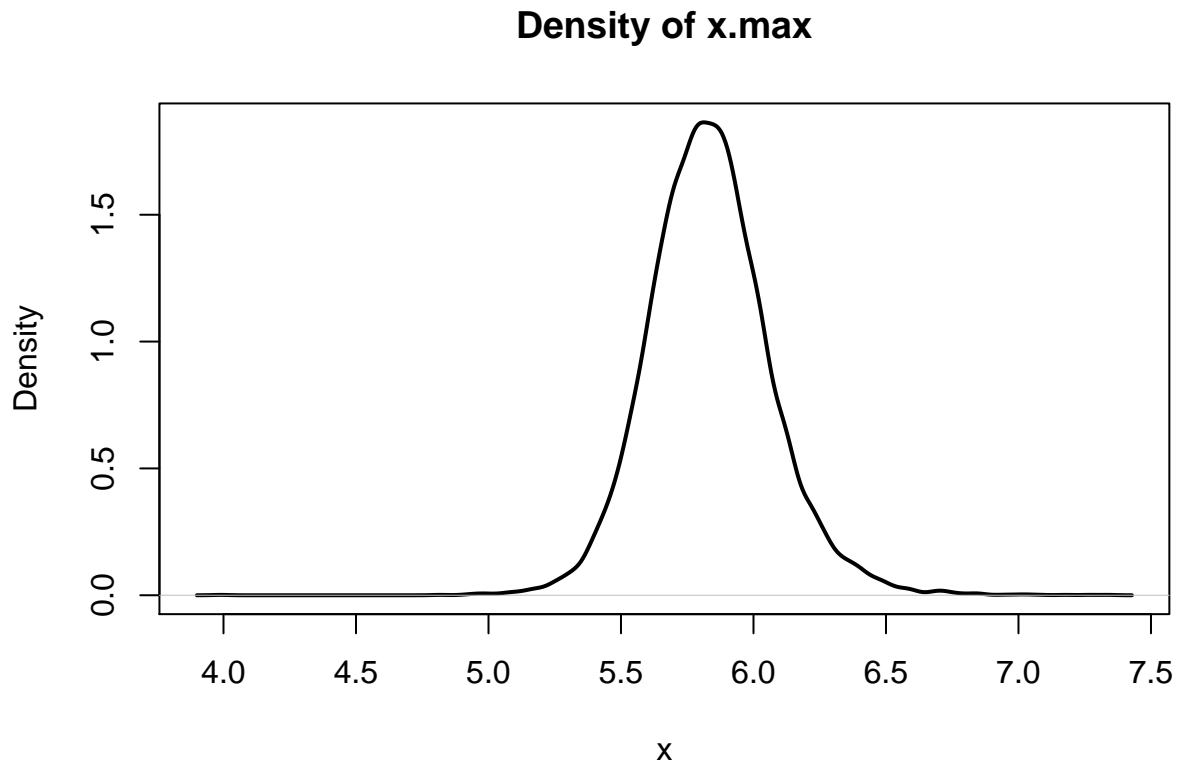
That the posterior densities of between plot variation of the intercept and slopes are maximized at values only slightly larger than zero suggest that there is some evidence of variation between groups, but that this variation could be stronger.

d.

i.

Since we are essentially estimating the coefficients of a quadratic equation( $c + bx + ax^2$ ), the value of  $x$  which maximizes the function will be  $\frac{-b}{2a}$ , or in this case  $x_{max} = \frac{-\beta_2}{2\beta_3}$ .

```
x.max = -mu.chain[,2]/(2*mu.chain[,3])
plot(density(x.max), lwd = 2, main = "Density of x.max", xlab = "x")
```



```
quantile(x.max, c(0.025,0.975))
```

```
##      2.5%      97.5%
## 5.408375 6.325960
```

A 95% Posterior CI for  $x_{max}$  would be [5.4, 6.3]

ii.

```
mean(x.max)
```

```
## [1] 5.833802
```

$\hat{x}_{max} = 5.83$

iii.

```
pe_x.max<-mean(x.max)
x.max.test<-c(1,mean(x.max), mean(x.max)^2)
beta.tilde<-matrix(NA, T, p)
for(t in 1:T){
  Sigma.post = Sigma.chain[t,]
  Sigma.post = matrix(Sigma.post, nrow = 3, byrow = T)
  beta.tilde[t,] = mvrnorm(1,mu.chain[t,], Sigma.post)
}
y.tilde = beta.tilde %**% x.max.test
quantile(y.tilde,c(0.025, 0.975))
```



```
##          2.5%          97.5%  
##  9.397193 16.279225
```

A 95% PPD CI for a plot with  $\hat{x}_{max}$  density is  $[9.4, 16.3]$