HW4 Solution

Jitong

2023-03-28

Question 1

a)

```
data_p1 = read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/interexp.dat", header = TRUE)
```

```
theta_ya = mean(data_p1$yA, na.rm=TRUE)
theta_yb = mean(data_p1$yB, na.rm=TRUE)
sigma2_ya = var(data_p1$yA, na.rm=TRUE)
sigma2_yb = var(data_p1$yB, na.rm=TRUE)
rho = cor(data_p1$yA, data_p1$yB, use="complete.obs")
print(paste("Mean of yA:", theta_ya))
```

```
## [1] "Mean of yA: 24.200487804878"
```

```
print(paste("Mean of yB:", theta_yb))
```

```
## [1] "Mean of yB: 24.8053488372093"
```

```
print(paste("Variance of yA:", sigma2_ya))
```

```
## [1] "Variance of yA: 4.09279975609756"
```

```
print(paste("Variance of yB:", sigma2_yb))
```

```
## [1] "Variance of yB: 4.69157785160576"
```

```
print(paste("Correlation between yA and yB:", rho))
```

```
## [1] "Correlation between yA and yB: 0.616450901318467"
```

b)

```
# Impute missing values with given algorithm
imp_data = data_p1
imp_data[is.na(imp_data[,1]),1] <- theta_ya + (imp_data[is.na(imp_data[,1]),2]-theta_yb)*rho*sqrt(sigma
imp_data[is.na(imp_data[,2]),2] <- theta_yb + (imp_data[is.na(imp_data[,2]),1]-theta_ya)*rho*sqrt(sigma
t.test(imp_data$yA, imp_data$yB, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  imp_data$yA and imp_data$yB
## t = -3.2807, df = 57, p-value = 0.00177
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.9850730 -0.2383347
## sample estimates:
## mean difference
##      -0.6117038
```

The result rejects the null hypothesis.

c)

```
### Gibbs sampler
THETA <- matrix(0, nrow = 1000, ncol = 2)
SIGMA <- matrix(0, nrow = 1000, ncol = 4)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(MCMCpack) # for the riwish function
```

```
## Loading required package: coda
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2023 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
set.seed(12)
## prior parameters
X = data_p1
p = 2
n = dim(X)[1]
mu0 <- c(theta_ya, theta_yb)
sd0 <- c(sqrt(sigma2_ya), sqrt(sigma2_yb))
cov_yayb = rho*sqrt(sigma2_ya)*sqrt(sigma2_yb)
L0 <- matrix(c(sigma2_ya, cov_yayb, cov_yayb, sigma2_yb), 2, 2)
nu0 <- p+2
S0 <- L0
###
### starting values
Sigma <- S0
X.full <- X
for(j in 1:p){
  X.full[is.na(X.full[,j]),j] <- mean(X.full[,j],na.rm=TRUE)
}
###
for(s in 1:1000){
  ## update theta
  xbar <- apply(X.full,2,mean)
```

```
  Ln <- solve(solve(L0) + n * solve(Sigma))
  mun <- Ln %*% (solve(L0) %*% mu0 + n * solve(Sigma) %*% xbar)
  theta <- mvrnorm(1, mun, Ln)

  ## update Sigma
  Sn <- S0 + (t(X.full) - c(theta)) %*% t(t(X.full) - c(theta))
  Sigma <- riwish(nu0 + n, Sn)

  ## update missing data
  for(i in 1:n){
    if(any(is.na(X[i,]))){
      index1 = which(is.na(X[i,]))
      index2 = which(!is.na(X[i,]))
      iS <- solve(Sigma[index2,index2])
      beta.j <- Sigma[index1,index2]%*%iS
      s2.j <- Sigma[index1,index1] - Sigma[index1,index2]%*%iS%*%Sigma[index2,index1]
      theta.j<- theta[index1] + beta.j%*%(t(X.full[i,index2])-theta[index2])
      X.full[i,index1] <- mvrnorm(1,theta.j,s2.j )
    }
  }
  # Update the output
  THETA[s,] <- theta
  SIGMA[s,] <- c(Sigma) # notice the Sigma matrix is vectorized
}

## Posterior mean
apply(THETA,2,mean)
```

```
## [1] 24.21150 24.81806
```

```
## Posterior CI
quantile(THETA[,1] - THETA[,2], probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -1.2810790  0.0250168
```

```
## Posterior difference
mean(THETA[,1] - THETA[,2])
```

```
## [1] -0.606558
```

The results fail to reject the null hypothesis. The Gibbs sampler is a better approach than imputing using the correlation.

Question 2

```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
```

```
##     cbind, rbind
```

```
init = mice(data_p1, maxit = 0)
meth = init$method
predM = init$predictorMatrix
set.seed(12)
imputed = mice(data_p1, method = meth, m = 4, maxit = 1000, printFlag = FALSE)
fit <- with(data = imputed, exp = glm(yA - yB ~ 1))
```

```
summary(pool(fit), m = 4, conf.int = TRUE)
```

```
##          term   estimate std.error statistic       df   p.value     2.5 %
## 1 (Intercept) -0.6171552 0.3835647 -1.608999 7.131478 0.1508629 -1.520763
##      97.5 %
## 1 0.2864526
```

The result fail to reject the null hypothesis which shows similar confidence interval with the Gibbs sampler results.

Question 3

a)

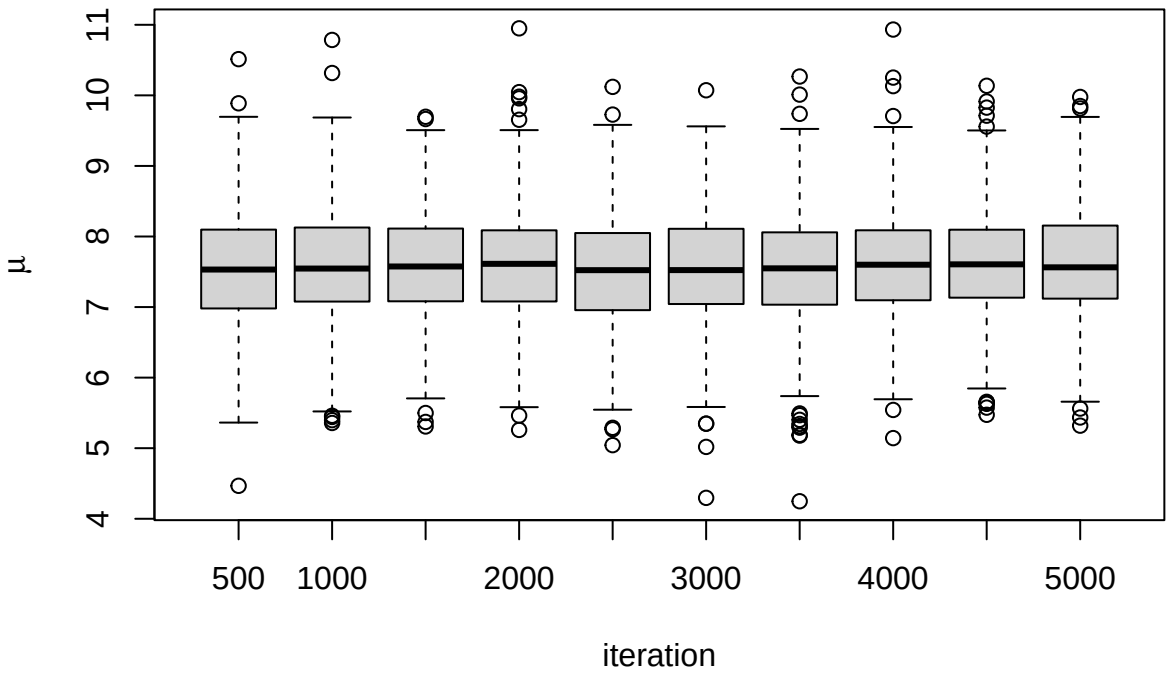
```
Y.school.score = NULL
for(i in 1:8){
  tmp = scan(paste0("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/school", i, ".dat"))
  tmp = cbind(i, tmp)
  Y.school.score = rbind(Y.school.score, tmp)
}
colnames(Y.school.score) = c('school', 'score')
```

```
Y <- list()
YM <- Y.school.score
J <- 8
n <- ybar <- ymed <- s2 <- rep(0,J)
for(j in 1:J){
  Y[[j]] <- Y.school.score[Y.school.score[,1] == j, 2]
  ybar[j] <- mean(Y[[j]])
  ymed[j] <- median(Y[[j]])
  n[j] <- length(Y[[j]])
  s2[j] <- var(Y[[j]])
}
## priors
nu0 <- 2
s20 <- 5
eta0 <- 2
t20 <- 10
mu0 <- 7
g20 <- 5
## starting values
N = sum(n)
theta <- ybar
sigma2 <- mean(s2)
mu <- mean(theta)
tau2 <- var(theta)
## setup MCMC
```

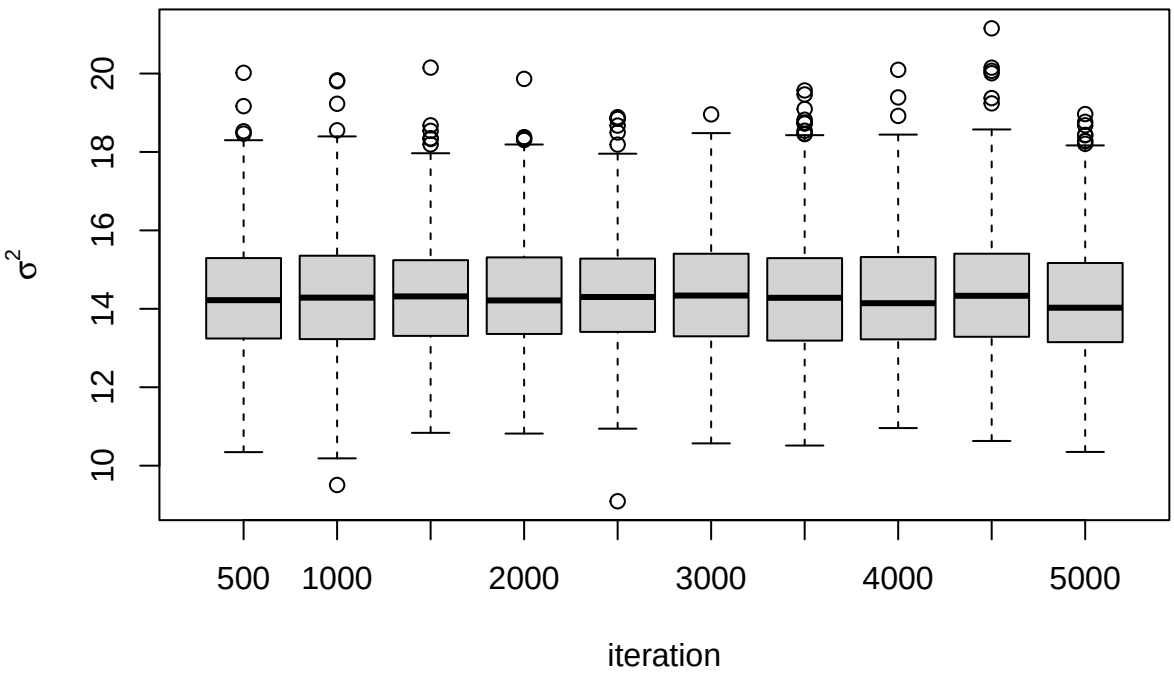
```r
set.seed(12)
S <- 5000
THETA <- matrix(0, nrow=S, ncol=J)
MST <- matrix(0, nrow=S, ncol=3)
## MCMC algorithm
for(s in 1:S){
  # sample new values of the thetas
  for(j in 1:J){
    vtheta <- 1/(n[j]/sigma2 + 1/tau2)
    etheta <- vtheta*(ybar[j]*n[j]/sigma2 + mu/tau2)
    theta[j]<-rnorm(1,etheta,sqrt(vtheta))
  }
  #sample new value of sigma2 from IG distribution
  nun = nu0 + N
  ss <- nu0*s20
  for(j in 1:J){
    ss <- ss + sum((Y[[j]]-theta[j])^2)
  }
  sigma2 <- 1/rgamma(1, nun/2, ss/2)
  #sample a new value of mu
  vmu <- 1/(J/tau2 + 1/g20)
  emu <- vmu*(J*mean(theta)/tau2 + mu0/g20)
  mu <- rnorm(1,emu,sqrt(vmu))
  # sample a new value of tau2
  etam <- eta0 + J
  ss<- eta0*t20 + sum((theta-mu)^2)
  tau2 <- 1/rgamma(1, etam/2, ss/2)
  #store results
  THETA[s,] <- theta
  MST[s,] <- c(mu, sigma2, tau2)
}
mcmc1 <- list(THETA=THETA, MST=MST)
```

Assess the convergence of parameters.

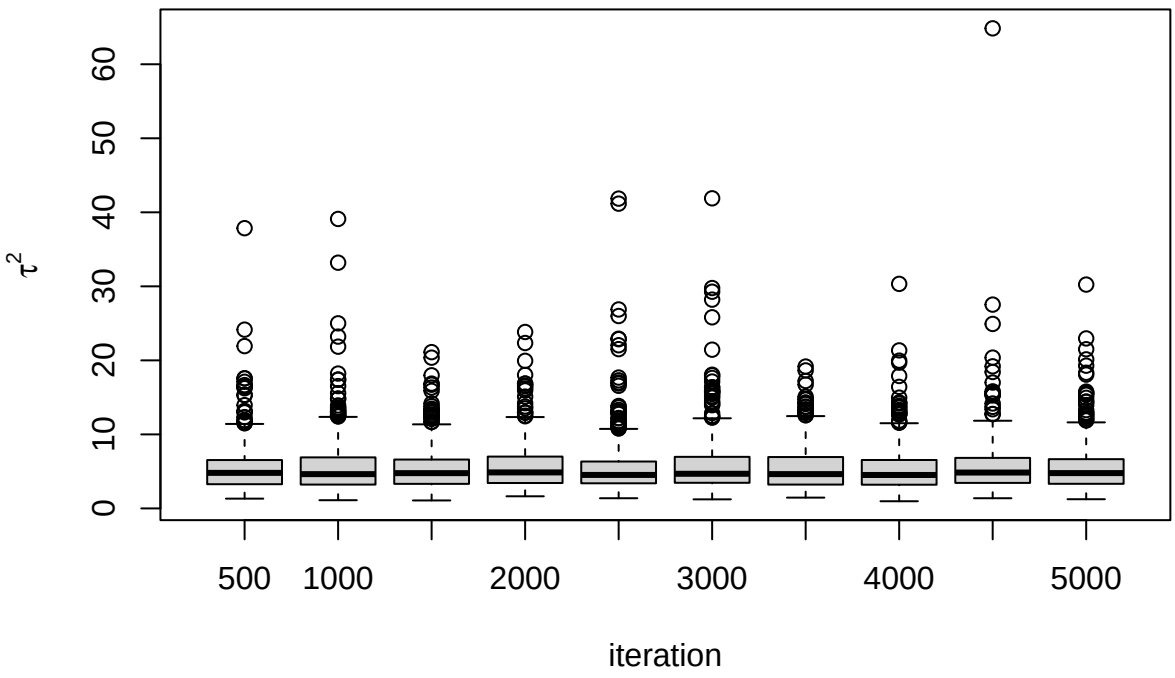
```r
stationarity.plot <- function(x,...){
  S = length(x)
  scan <- 1:S
  ng <- min(round(S/100), 10)
  group <- S * ceiling(ng*scan/S) / ng
  boxplot(x ~ group,...)
}

stationarity.plot(MST[,1],xlab="iteration",ylab=expression(mu))
```




```
stationarity.plot(MST[,2],xlab="iteration",ylab=expression(sigma^2))
```




```
stationarity.plot(MST[,3],xlab="iteration",ylab=expression(tau^2))
```



Check effective size.

```
library(coda)
effectiveSize(MST)
```

```
##     var1     var2     var3
## 4219.616 4647.827 3512.719
```

b)

```
post_mu = MST[,1]
post_sig2 = MST[,2]
post_tau2 = MST[,3]
print(paste("The posterior mean for mu is:", mean(post_mu)))
```

```
## [1] "The posterior mean for mu is: 7.56318340448127"
```

```
print(paste("The posterior mean for sigma^2 is:", mean(post_sig2)))
```

```
## [1] "The posterior mean for sigma^2 is: 14.3490460204683"
```

```
print(paste("The posterior mean for tau^2 is:", mean(post_tau2)))
```

```
## [1] "The posterior mean for tau^2 is: 5.5649984891326"
```

```
print("The posterior 95% CI for mu is:")
```

```
## [1] "The posterior 95% CI for mu is:"
```

```
quantile(post_mu, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 5.964291 9.099429
```

```
print("The posterior 95% CI for sigma^2 is:")
```

```
## [1] "The posterior 95% CI for sigma^2 is:"
```

```
quantile(post_sig2, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 11.64501 17.76382
```

```
print("The posterior 95% CI for tau^2 is:")
```

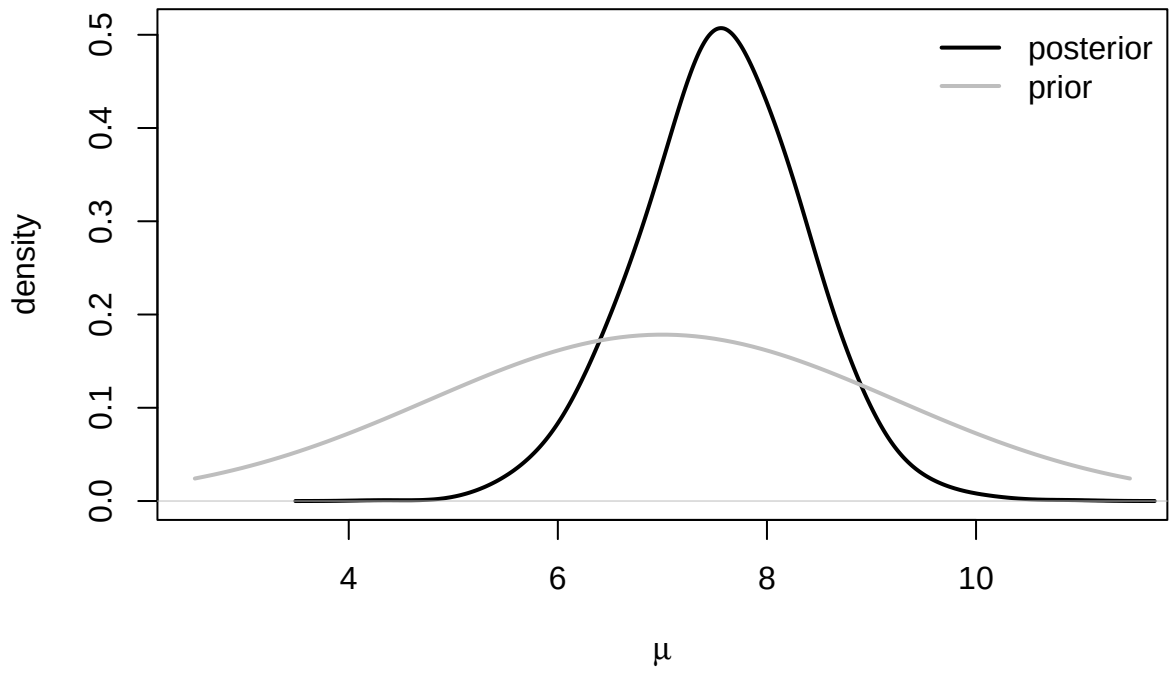
```
## [1] "The posterior 95% CI for tau^2 is:"
```

```
quantile(post_tau2, probs = c(0.025, 0.975))
```

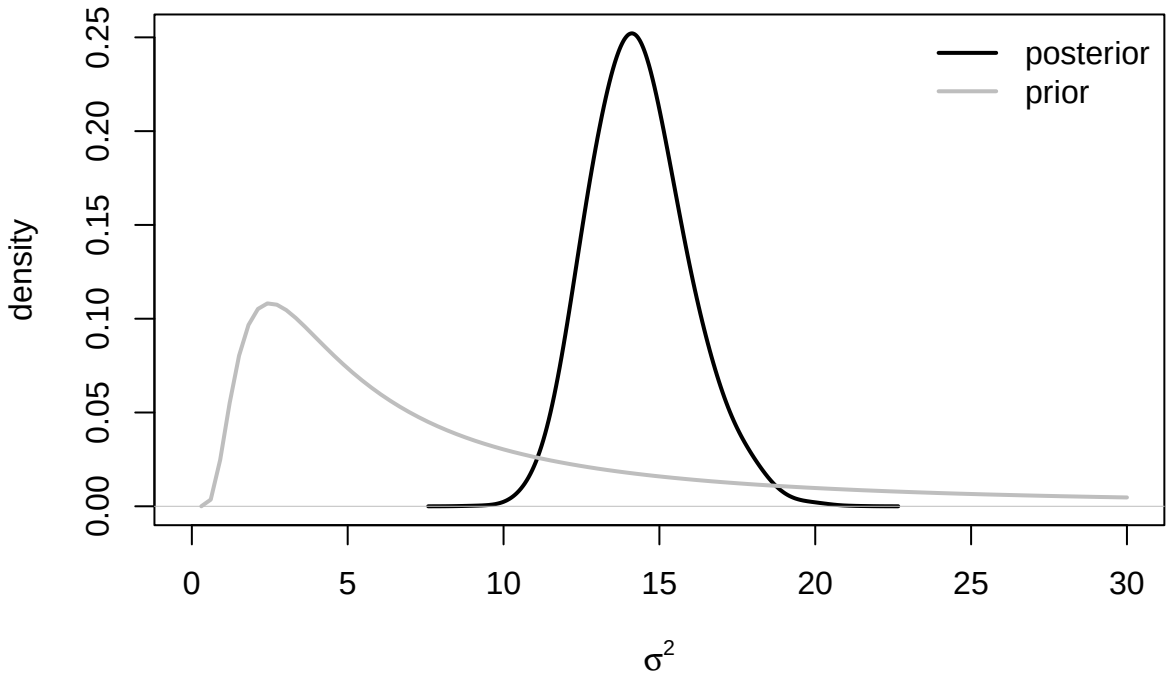
```
##      2.5%     97.5%
##  1.881417 14.163134
```

Compare the posterior and prior densities.

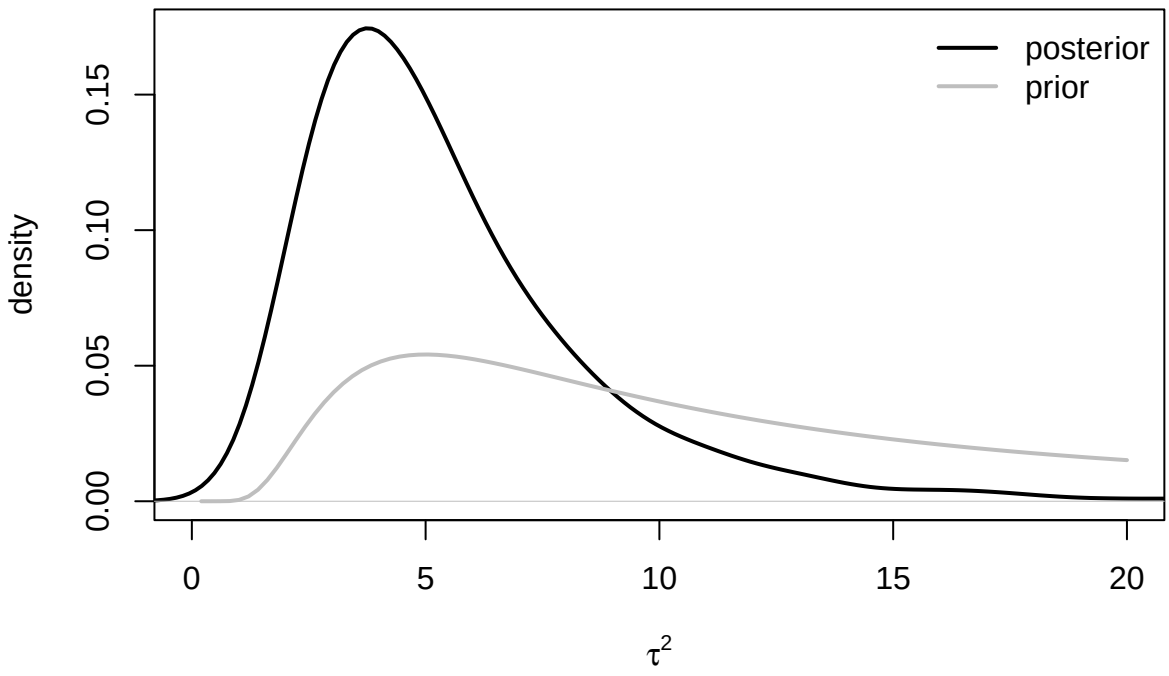
```
par(ask=F)
plot(density(MST[,1],adj=2), xlim = c(mu0-2*sqrt(g20), mu0+2*sqrt(g20)), main="", xlab=expression(mu), y
ds <- seq(mu0-2*sqrt(g20), mu0+2*sqrt(g20), length=100)
lines(ds, dnorm(ds,mu0,sqrt(g20)),lwd=2,col="gray" )
legend("topright",legend=c("posterior","prior"),lwd=c(2,2),col=c("black","gray"), bty="n")
```



```r
par(ask=F)
plot(density(MST[,2],adj=2), xlim=c(0,30), main="", xlab=expression(sigma^2),ylab="density",lwd=2)
ds <- seq(0,30,length=100)
lines(ds, dinvgamma(ds,nu0/2,nu0/2*s20),lwd=2,col="gray")
legend("topright",legend=c("posterior","prior"),lwd=c(2,2),col=c("black","gray"), bty="n")
```

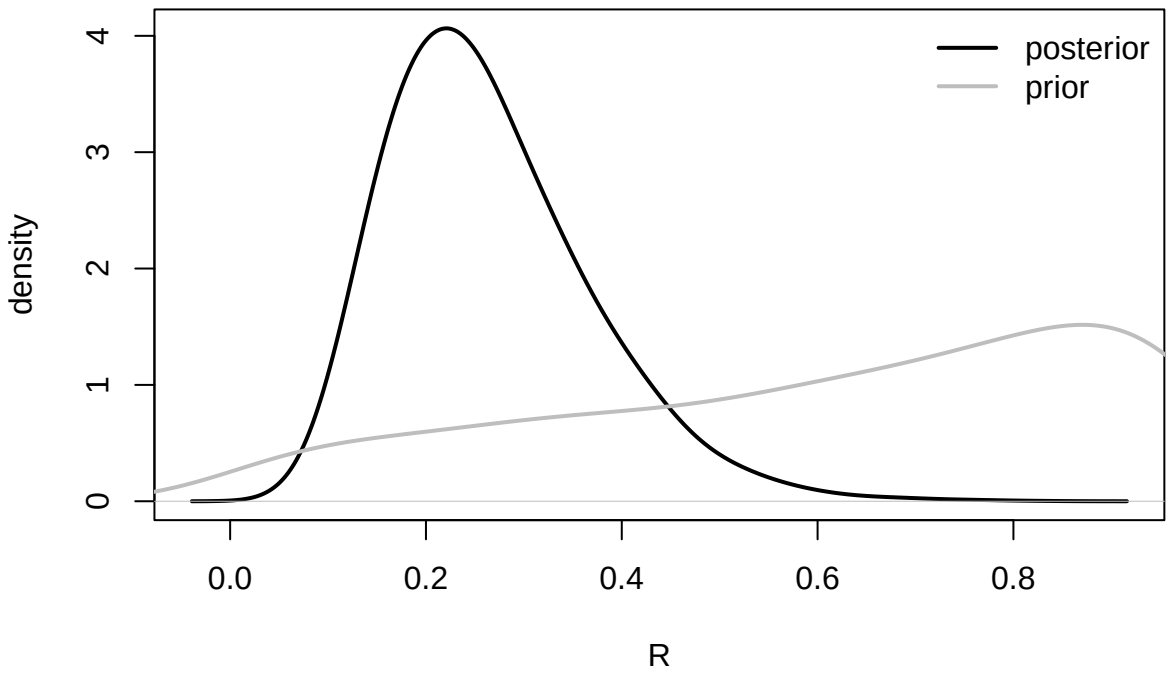



```r
par(ask=F)
plot(density(MST[,3],adj=2),xlim=c(0,20), main="", xlab=expression(tau^2),ylab="density", lwd=2)
ds<-seq(0,20,length=100)
lines(ds,dinvgamma(ds,eta0/2,eta0/2*t20),lwd=2,col="gray")
legend("topright",legend=c("posterior","prior"),lwd=c(2,2),col=c("black","gray"), bty="n")
```



c)

```
par(ask = F)
tau2_tmp = MST[,3]
sig2_tmp = MST[,2]
post_R = tau2_tmp / (tau2_tmp+sig2_tmp)
plot(density(post_R, adj=2), main="",xlab=expression(R),ylab="density",lwd=2, col ='black')
pri_tau2 = rinvgamma(10000, eta0/2,eta0/2*t20)
pri_sig2 = rinvgamma(10000, nu0/2,nu0/2*s20)
pri_R = pri_tau2 / (pri_tau2+pri_sig2)
lines(density(pri_R, adj=2), main="",xlab=expression(R),ylab="density",lwd=2, col ='grey')
legend("topright",legend=c("posterior","prior"),lwd=c(2,2),col=c("black","gray"), bty="n")
```



The posterior mode is close to 0.2 indicating the variation between groups accounts for around 20% of the total variation.

d)

```
theta_7 = THETA[,7]
theta_6 = THETA[,6]
mean(theta_7 < theta_6)
```

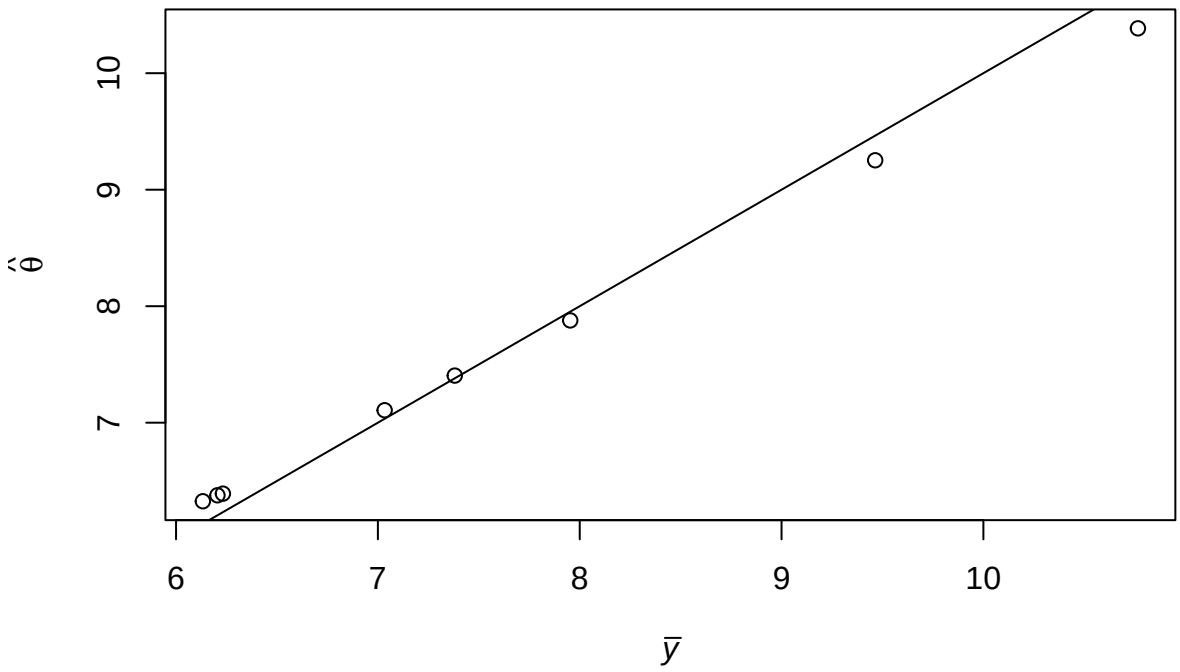
```
## [1] 0.5196
```

```
min_theta = apply(THETA, 1, FUN = min)
mean(theta_7 == min_theta)
```

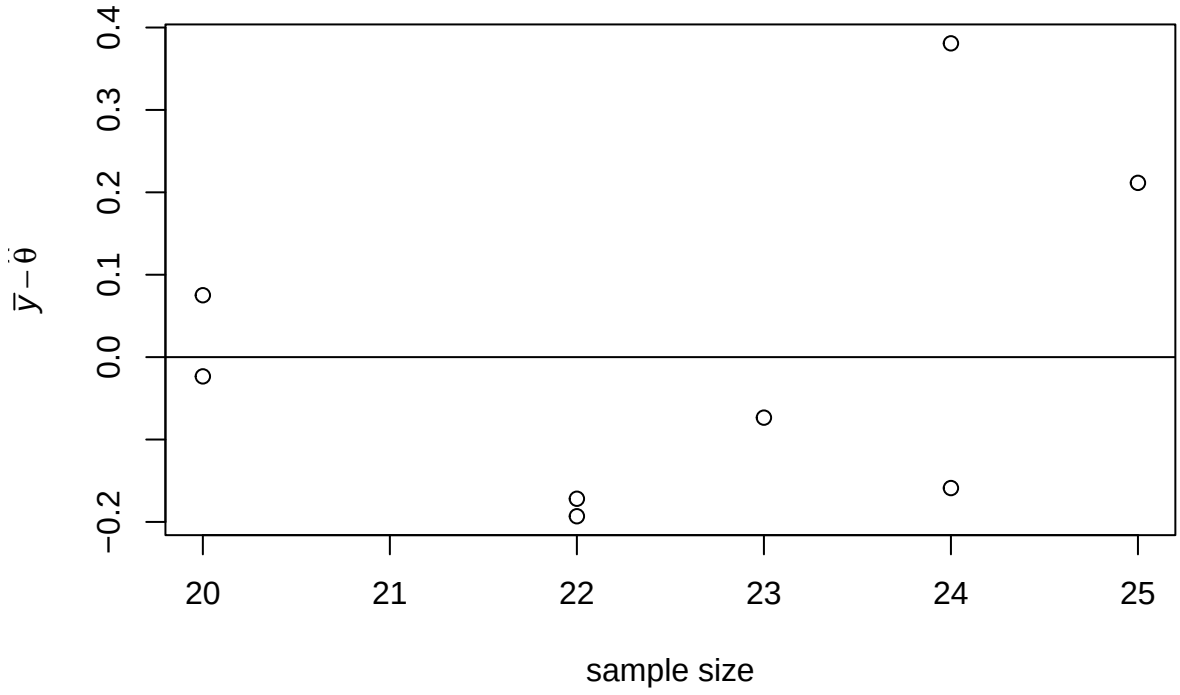
```
## [1] 0.3272
```

e)

```
theta.hat<-apply(THETA,2,mean)
plot(ybar,theta.hat,xlab=expression(bar(italic(y))),ylab=expression(hat(theta)))
abline(0,1)
```




```
plot(n,ybar-theta.hat,ylab=expression( bar(italic(y))-hat(theta) ),xlab="sample size")
abline(h=0)
```




```
sample.mean = mean(Y.school.score[,2])
print(paste("Sample Mean:", sample.mean))
```

```
## [1] "Sample Mean: 7.69127777777778"
```

```
post.mean.mu = mean(MST[,1])
print(paste("Post Mean of mu:", post.mean.mu))
```

```
## [1] "Post Mean of mu: 7.56318340448127"
```

The posterior mean is smaller since the prior mean is set as 7.

Question 4

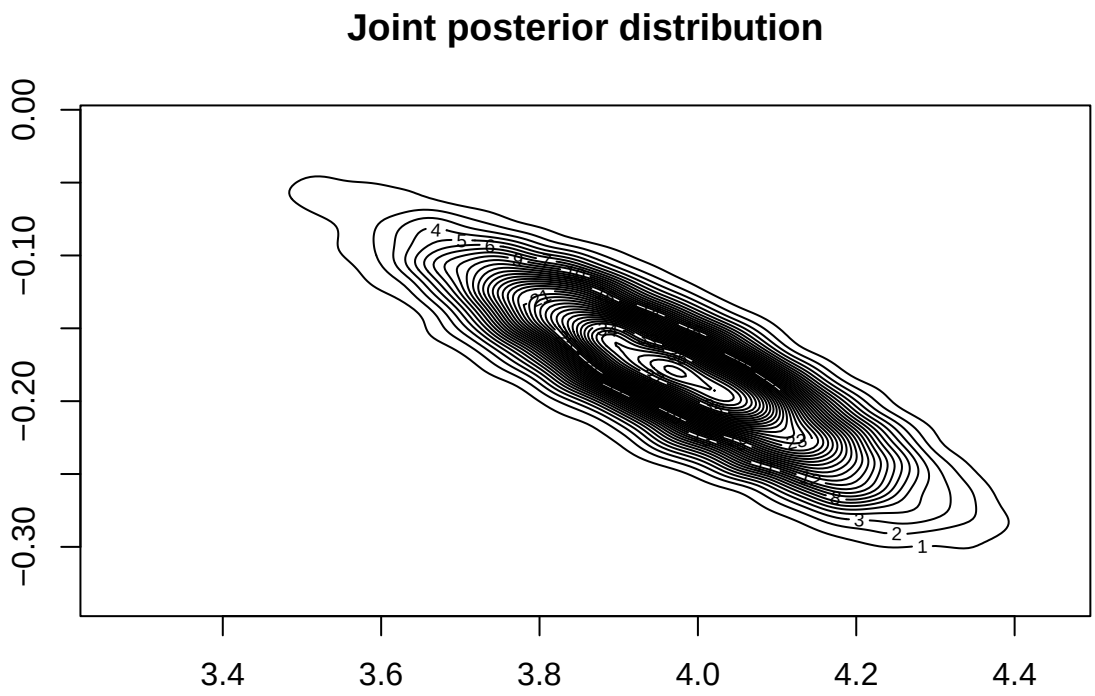
```
library(rstan)
write(
"data {
  int<lower=0> N;
  int<lower=0> x[N];
}
parameters {
  real alpha;
  real beta;
}
transformed parameters {
  real x7_exp = exp(alpha+7*beta);
}
```

```
model {
  for (i in 1:N)
    x[i] ~ poisson(exp(alpha+i*beta));
  // prior
  alpha ~ normal(0, 10);
  beta ~ normal(0, 10);
}
",
"p4.stan")
model = stan_model("p4.stan")
data = list(N = 6, x = c(64, 13, 33, 18, 30, 20))
fit = sampling(model, data, iter = 5000, chains = 4)
```

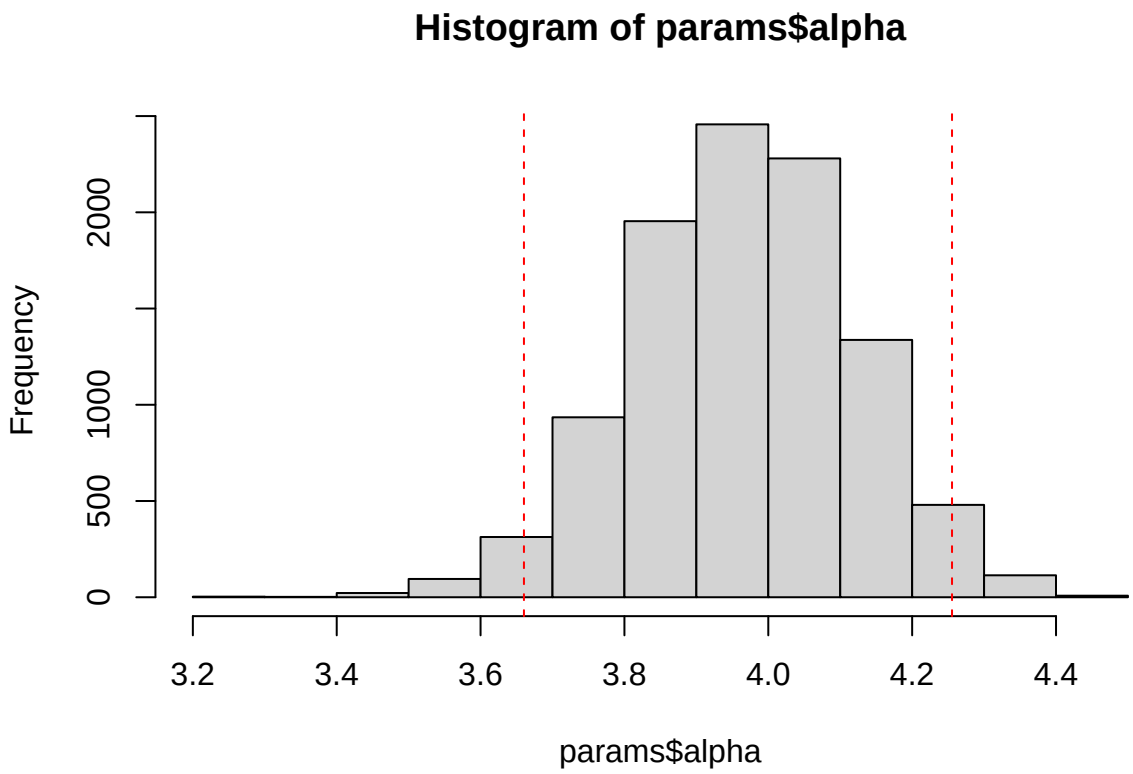
```
print(fit)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
##          mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha    3.97    0.00 0.15   3.66   3.86   3.97   4.07   4.26  2383    1
## beta    -0.18    0.00 0.05  -0.27  -0.21  -0.18  -0.15  -0.09  2362    1
## x7_exp  15.31    0.05 3.02  10.20  13.17  15.05  17.15  21.92  3021    1
## lp__   432.58    0.02 1.01 429.78 432.21 432.88 433.29 433.55  2837    1
##
## Samples were drawn using NUTS(diag_e) at Thu Mar 30 14:58:47 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

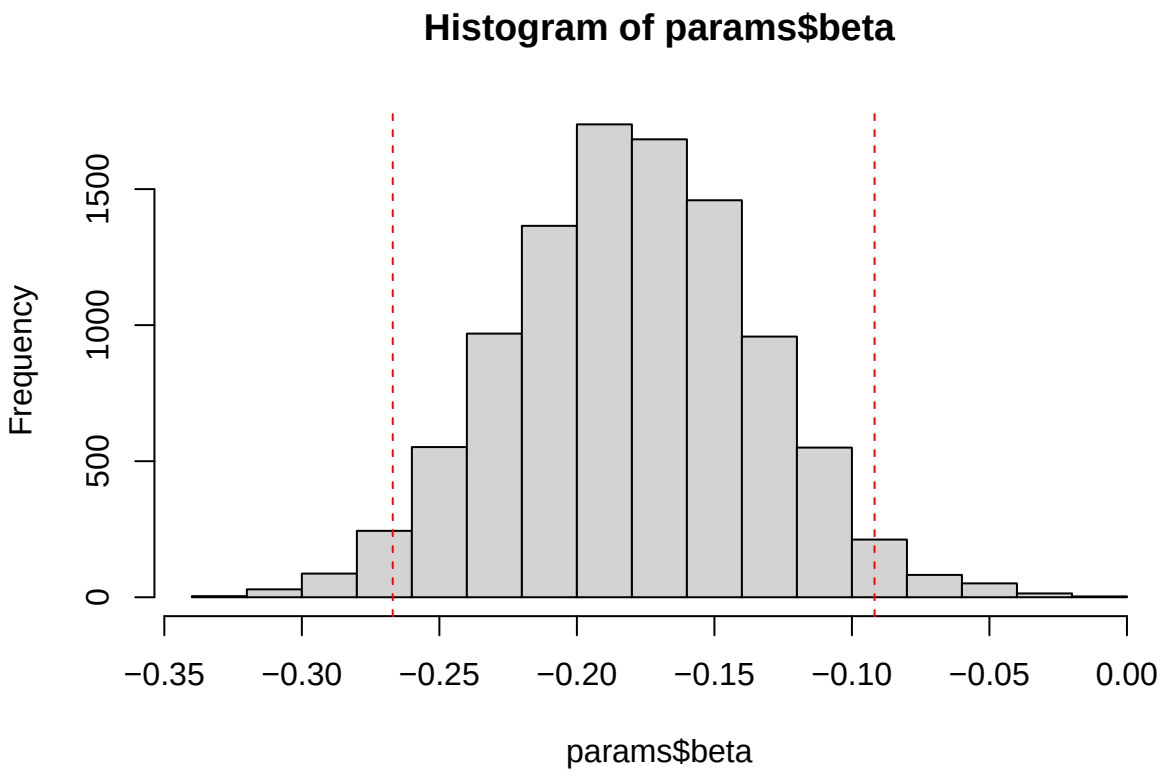
```
params = extract(fit)
bivn.kde <- kde2d(params$alpha, params$beta, n = 200)
contour(bivn.kde, nlevels = 50)
title("Joint posterior distribution")
```




```
hist(params$alpha)
abline(v = quantile(probs = c(0.025), params$alpha), col = "red", lty= 2)
abline(v = quantile(probs = c(0.975), params$alpha), col = "red", lty= 2)
```




```
hist(params$beta)
abline(v = quantile(probs = c(0.025), params$beta), col = "red", lty= 2)
abline(v = quantile(probs = c(0.975), params$beta), col = "red", lty= 2)
```



b)

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

```
##       2.5%      97.5%
## -0.26697245 -0.09178757
```

c)

```
quantile(params$x7_exp, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 10.20146 21.91677
```

d)

```
ALPHA = params$alpha
BETA = params$beta
PRED_x7 = rep(0, 10000)
for(i in 1:10000){
  PRED_x7[i] = rpois(1, exp(ALPHA[i] + 7*BETA[i]))
}
quantile(PRED_x7, c(0.025, 0.975))
```

```
##  2.5% 97.5%
##     7    26
```