HW4

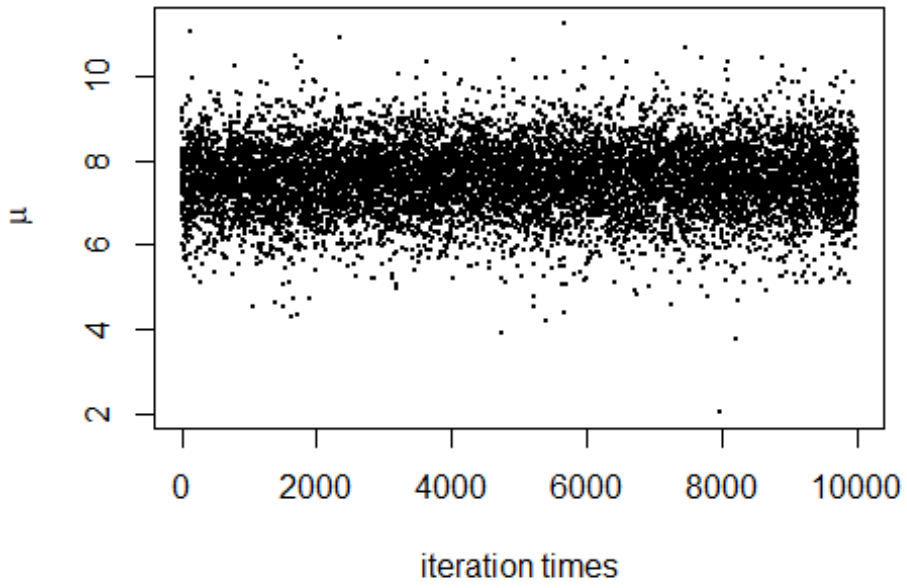
Guanren Wang

2019-4-4

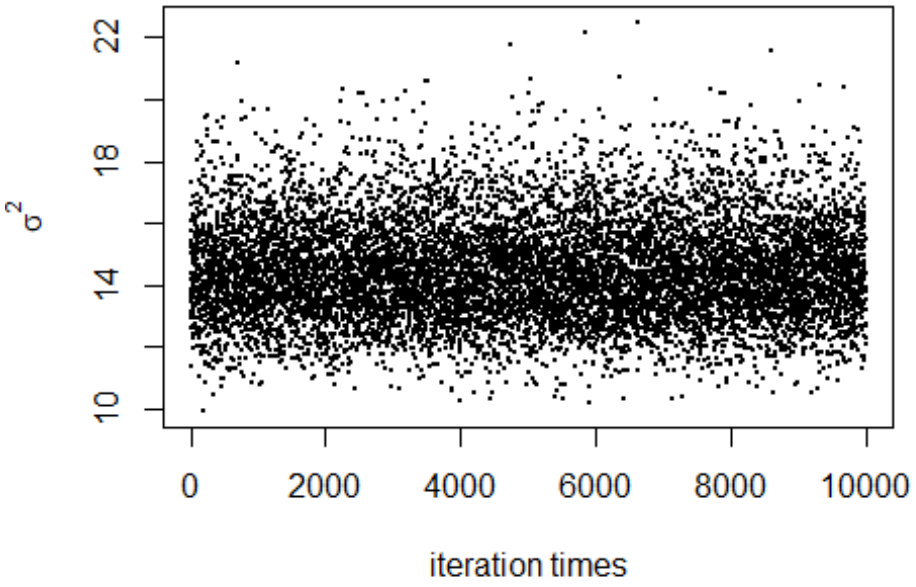
1.(a).

```
#load files
library(tidyverse)
wd<-'./school'
school<-list()
for (i in 1:8) {
  datapath<-paste(wd,i,'.txt',sep = '')
  school[[i]]<-read.table(datapath,header = F)[,1]
}

#set priors
mu0=7
gamma0_2=5
tau0_2=10
eta0=2
sigma0_2=15
nu0=2

#set starting values
s=10000
m=length(school)
theta<-matrix( nrow=s , ncol=m)
variance<-c()

for (i in 1:m) {
  theta[1,i]<-mean(school[[i]])
  variance[i]<-var(school[[i]])
}
mu<-mean(theta[1,])
tau_2<-var(theta[1,])
sigma_2<-mean(variance)
n<-lapply(school, length)

#start gibbs sampler
set.seed(123)
for (i in 2:s) {
  mu[i]=rnorm(n=1,
              mean=(m*mean(theta[i-1,])/tau_2[i-1]+mu0/gamma0_2)/(m/tau
_2[i-1]+1/gamma0_2),
              sd=sqrt(1/(m/tau_2[i-1]+1/gamma0_2))
```

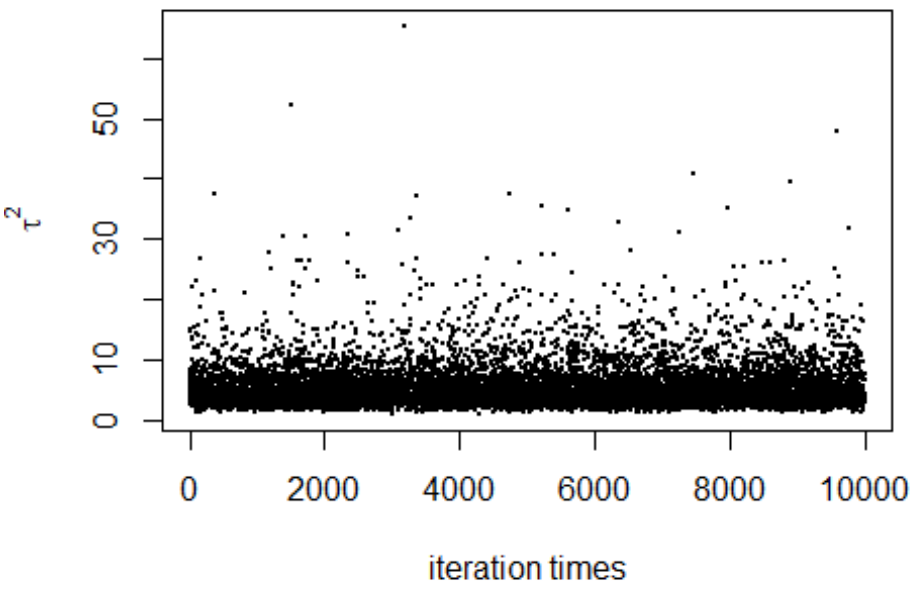
```
              )

  tau_2[i]=1/rgamma(n=1,
                    shape=(eta0+m)/2,
                    rate=(eta0*tau0_2+sum((theta[i-1,]-mu[i])^2))/2
                    )

  standard=0 #sum of squares
  for (j in 1:m) {
    a<-sum((school[[j]]-theta[i-1,j])^2)
    standard<-a+standard
  }
  sigma_2[i]=1/rgamma(n=1,shape=(nu0+sum(unlist(n)))/2,rate=(nu0*sigma0
_2+standard)/2)

  for (j in 1:m) {
    theta[i,j]=rnorm(n=1,
                     mean=(unlist(n)[j]*mean(school[[j]])/sigma_2[i]+mu
[i]/tau_2[i])/ (unlist(n)[j]/sigma_2[i]+1/tau_2[i]),
                     sd=sqrt(1/(unlist(n)[j]/sigma_2[i]+1/tau_2[i])))
  }
}

#check stationarity
plot(1:s,mu,pch=20,xlab = 'iteration times',ylab = expression(mu),cex=0.
5)
```



```r
plot(1:s,sigma_2,pch=20,xlab = 'iteration times',ylab = expression(sigm
a^2),cex=0.5)
```




```r
plot(1:s,tau_2,pch=20,xlab = 'iteration times',ylab = expression(tau^2),
cex=0.5)
```




```
#calculate effective sample size
library(coda)
paste('Effective size of',expression(mu),'is',effectiveSize(mu))%>%prin
t()

## [1] "Effective size of mu is 8581.80272742388"

paste('Effective size of',expression(sigma^2),'is',effectiveSize(sigma_
2))%>%print()

## [1] "Effective size of sigma^2 is 9596.99411971756"

paste('Effective size of',expression(tau^2),'is',effectiveSize(tau_
2))%>%print()

## [1] "Effective size of tau^2 is 6767.06877004341"
```

Obviously, the convergence of MCMC for 3 posteriors is good.

1.(b).

```
paste('posterior mean of sigma^2 is',mean(sigma_2))%>%print()

## [1] "posterior mean of sigma^2 is 14.4829402132339"

paste('posterior mean of tau_2 is',mean(tau_2))%>%print()

## [1] "posterior mean of tau_2 is 5.55062013572763"
```

```r
paste('posterior mean of mu is',mean(mu))%>%print()
```

```
## [1] "posterior mean of mu is 7.55579145384913"
```

```r
paste(paste(c('Lower','Upper'),'bound 95% CI of sigma^2 is'),quantile(s
igma_2,c(0.025,0.975)))%>%print()
```

```
## [1] "Lower bound 95% CI of sigma^2 is 11.7539686182915"
## [2] "Upper bound 95% CI of sigma^2 is 18.0026073121562"
```

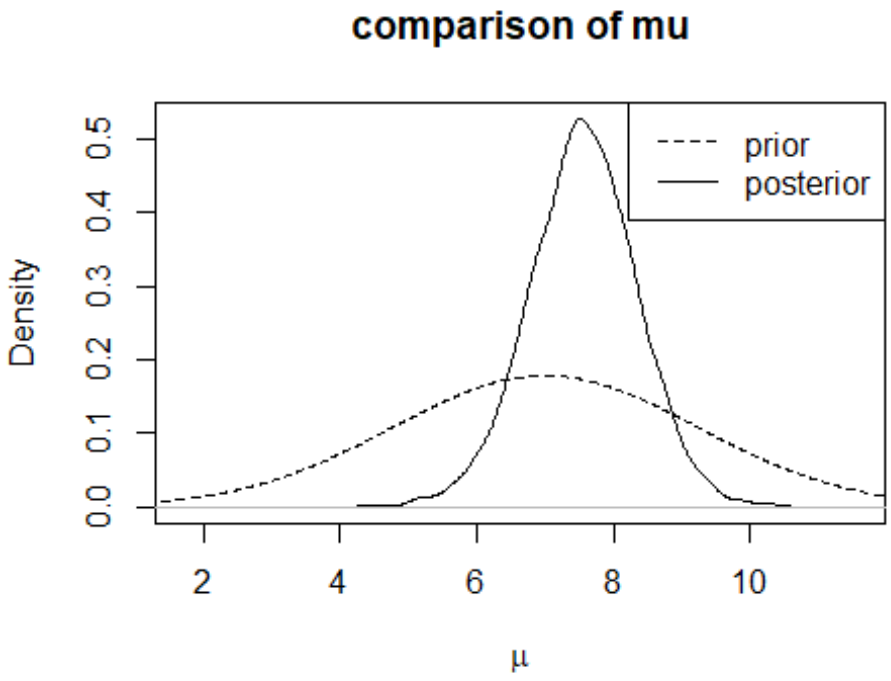
```r
paste(paste(c('Lower','Upper'),'bound 95% CI of tau^2 is'),quantile(tau
_2,c(0.025,0.975)))%>%print()
```

```
## [1] "Lower bound 95% CI of tau^2 is 1.90698973233749"
## [2] "Upper bound 95% CI of tau^2 is 15.1530186040079"
```

```r
paste(paste(c('Lower','Upper'),'bound 95% CI of mu is'),quantile(mu,c(0.
025,0.975)))%>%print()
```

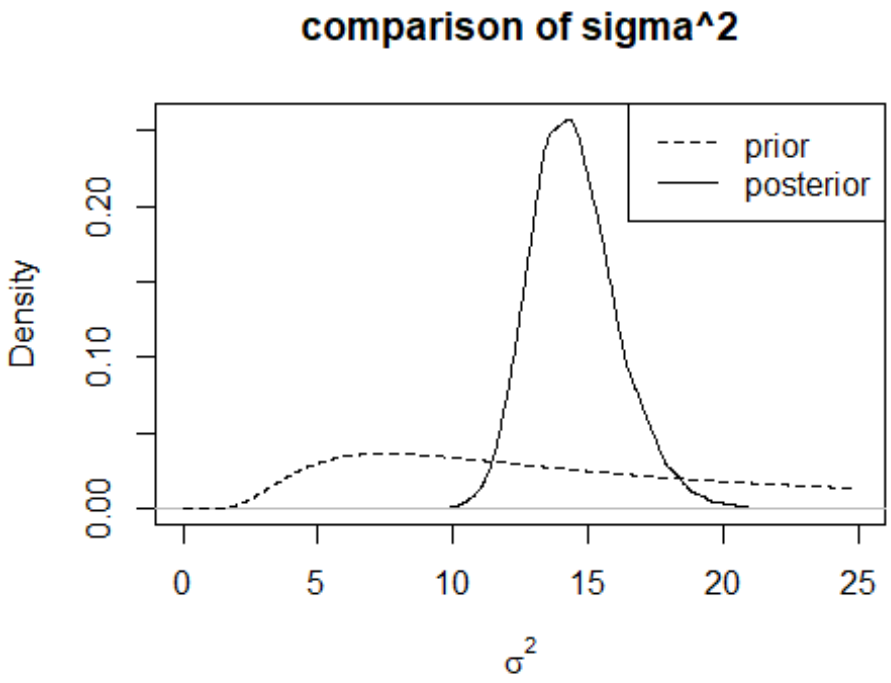
```
## [1] "Lower bound 95% CI of mu is 5.93994846363839"
## [2] "Upper bound 95% CI of mu is 9.12126629398086"
```

```r
grid=seq(0,20,0.01)
plot(density(mu),xlab = expression(mu),main = 'comparison of mu')
lines(grid,dnorm(x=grid,mean=mu0,sd=sqrt(gamma0_2)),type='l',lty=2)
legend('topright',c('prior','posterior'),lty=c(2, 1))
```

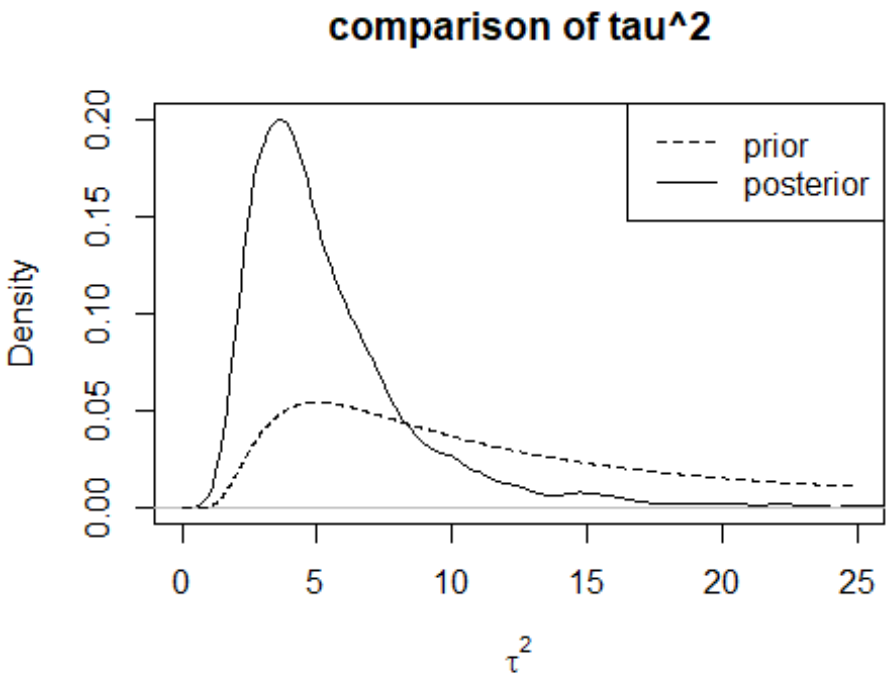



```r
library(nimble)
grid=seq(0,25,0.01)
```

```
plot(density(sigma_2),xlab = expression(sigma^2),main = 'comparison of
sigma^2',xlim=c(0,25))
lines(grid,dinvgamma(x=grid,shape=nu0/2,scale = nu0*sigma0_2/2),type='l
',lty=2)
legend('topright',c('prior','posterior'),lty=c(2, 1))
```



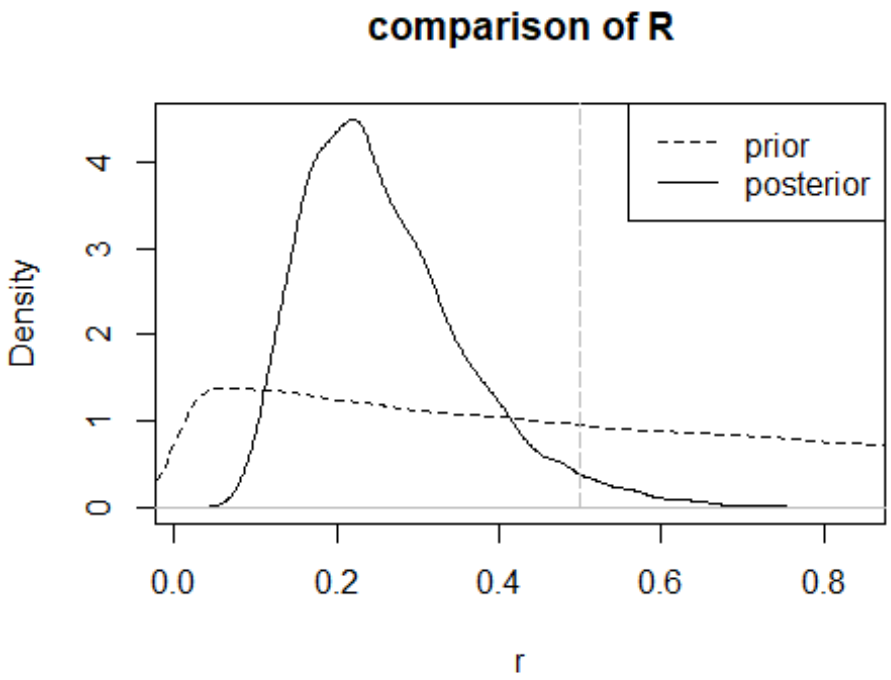

```
plot(density(tau_2),xlab = expression(tau^2),main = 'comparison of tau^
2',xlim=c(0,25))
lines(grid,dinvgamma(x=grid,shape=eta0/2,scale = eta0*tau0_2/2),type='l
',lty=2)
legend('topright',c('prior','posterior'),lty=c(2, 1))
```



Three prior distributions are all very flattened,which means they are uninformative. Plus, kurtosis of our posterior distributions are much bigger than prior distributions. Thus, we learned from data that μ,σ^2 and τ^2 should concentrate more on certain values.

1.(c).

```
r=tau_2/(tau_2+sigma_2)

s=100000
sig_prior<-1/rgamma(n=s,shape=nu0/2,rate = nu0*sigma0_2/2)
tau_prior<-1/rgamma(n=s,shape=eta0/2,rate = eta0*tau0_2/2)
R=tau_prior/(tau_prior+sig_prior)
plot(density(r),xlab = 'r',main = 'comparison of R')
lines(density(R),lty=2)
lines(x=rep(0.5,6),y=0:5,lty=5,col='grey')
legend('topright',c('prior','posterior'),lty=c(2, 1))
```



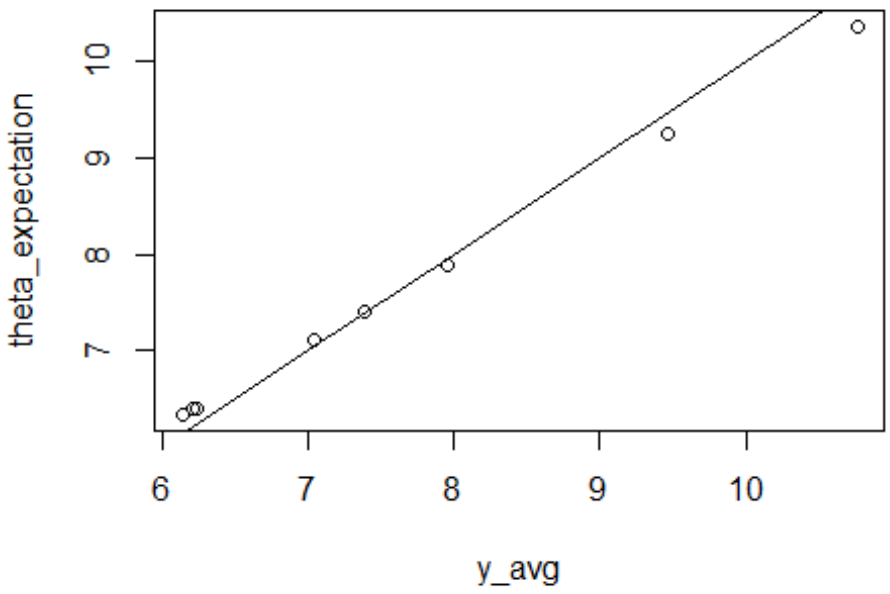
τ^2 represents the between group variation, while σ^2 represents within group variation. Thus R measures the ratio of between group variation to all the variation. From the posterior we could find most R is less than 0.5 (marked in solid line), indicating a larger proportion of between group variation relative to within group variation.

1.(d).

```
paste('The probability of theta7 is smaller than theta6 is',mean(theta[,
7]<theta[,6]))%>%print()

## [1] "The probability of theta7 is smaller than theta6 is 0.5248"

paste('The probability of theta7 is smallest of all thetas is',mean(the
ta[,7]==apply(theta,1,min)))%>%print()

## [1] "The probability of theta7 is smallest of all thetas is 0.3213"
```

1.(e).

```
theta_expectation<-apply(theta,2,mean)
y_avg<-lapply(school,mean)
plot(y_avg,theta_expectation)
lines(x=6:11,y=6:11)
```




```
mean(mu)

## [1] 7.555791

mean(unlist(school))

## [1] 7.691278
```

Notice that the relationship roughly follows a line with a slope that is less than one, indicating that high values of $\overline{y}$ correspond to slightly less high values of θ, and low values of $\overline{y}$ correspond to slightly less low values of θ. Expectation of θ is is pulled a bit from $\overline{y}$ towards μ by an amount depending on n_j.

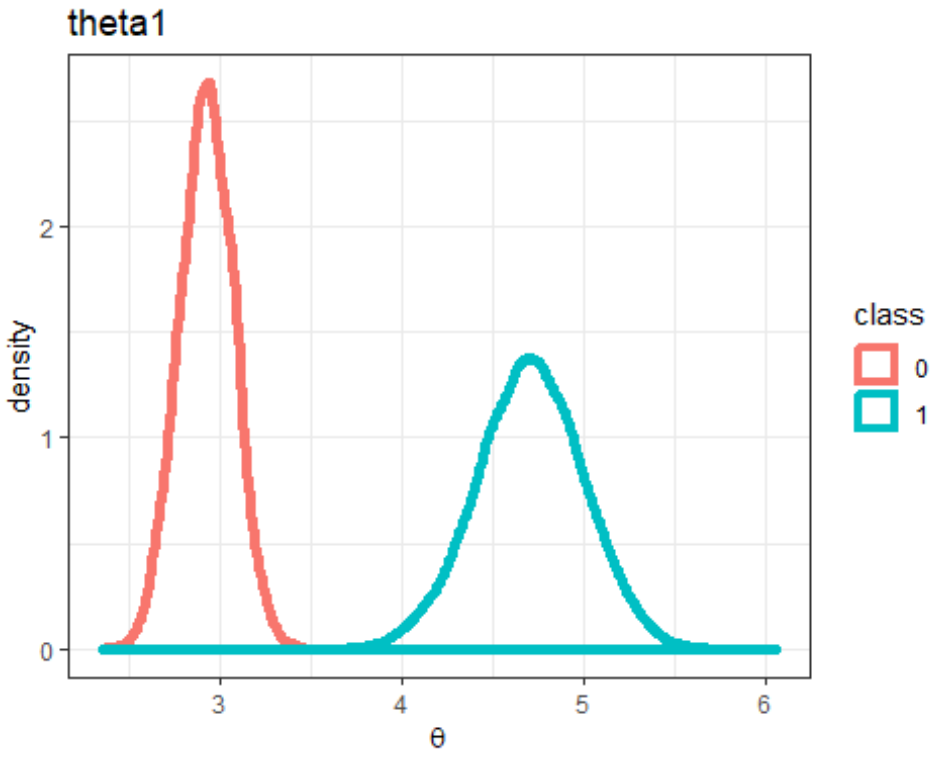
The mean of μ is very close to the average of all observations.

2.(a).

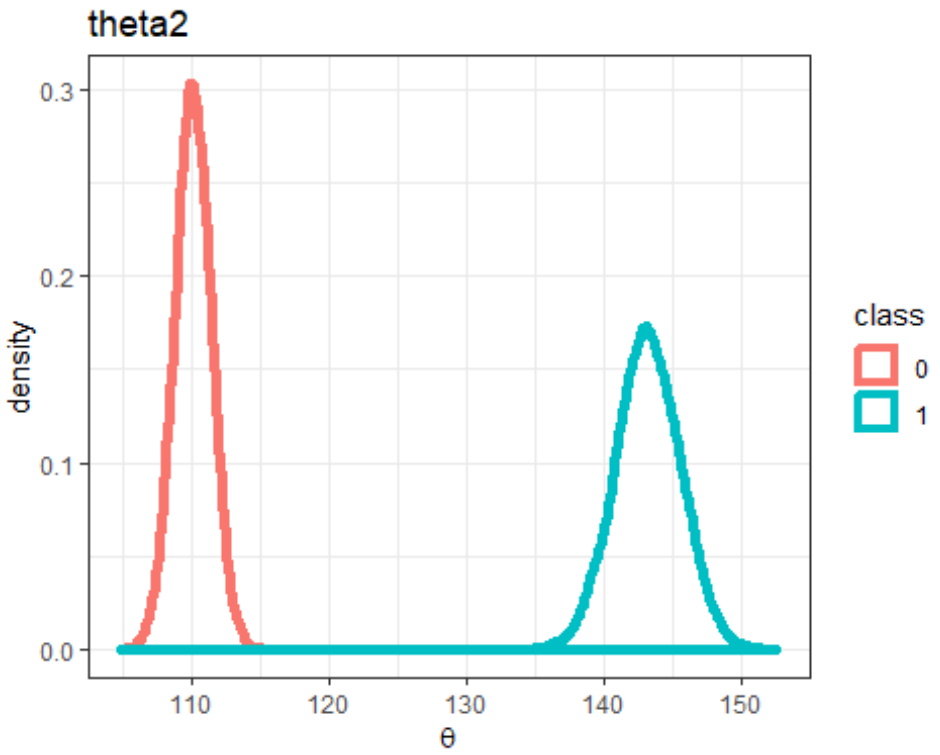
```
dt<-read.table('./azdiabetes.txt',header = T)
d<-as.matrix(dt[dt[,8]=='Yes',][,-8])
n<-as.matrix(dt[dt[,8]=='No',][,-8])

#set starting values
mu0_d=apply(d,2,mean)
mu0_n=apply(n,2,mean)
y.bar_d=mu0_d
y.bar_n=mu0_n
lambda0_d=s0_d=cov(d)
```

```
lambda0_n=s0_n=cov(n)
nu0_d=nu0_n=9
s=10000
n_d=dim(d)[1]
n_n=dim(n)[1]

#start sampling
theta_d=matrix(nrow = s,ncol = dim(d)[2])
theta_n=matrix(nrow = s,ncol = dim(n)[2])
sigma_d=array(dim = c(dim(s0_d),s))
sigma_n=array(dim = c(dim(s0_n),s))
theta_d[1,]=y.bar_d
theta_n[1,]=y.bar_n
sigma_d[,,1]=lambda0_d
sigma_n[,,1]=lambda0_n
library(Rfast)
library(monomvn)

set.seed(123)
for (i in 2:s) {
  #sample d
  lambdan_d=solve(solve(lambda0_d)+n_d*solve(sigma_d[,,i-1]))
  lambdan_d[upper.tri(lambdan_d)]=0
  lambdan_d=lambdan_d+t(lambdan_d)
  diag(lambdan_d)=diag(lambdan_d)/2
  mun_d=lambdan_d%*%(solve(lambda0_d)%*%mu0_d+n_d*solve(sigma_d[,,i-1])%
*%y.bar_d)
  theta_d[i,]=rmvnorm(n=1,mu=mun_d,sigma=lambdan_d)

  nun_d=nu0_d+n_d
  s_theta_d=0
  for (j in 1:n_d) {
    s_theta_d=s_theta_d+(d[j,]-theta_d[i,])%*%t(d[j,]-theta_d[i,])
  }
  sn_d=s0_d+s_theta_d
  sigma_d[,,i]=solve(rwish(v=nun_d,S=solve(sn_d)))

  #sample n
  lambdan_n=solve(solve(lambda0_n)+n_n*solve(sigma_n[,,i-1]))
  lambdan_n[upper.tri(lambdan_n)]=0
  lambdan_n=lambdan_n+t(lambdan_n)
  diag(lambdan_n)=diag(lambdan_n)/2

  mun_n=lambdan_n%*%(solve(lambda0_n)%*%mu0_n+n_n*solve(sigma_n[,,i-1])%
*%y.bar_n)
  theta_n[i,]=rmvnorm(n=1,mu=mun_n,sigma=lambdan_n)

  nun_n=nu0_n+n_n
  s_theta_n=0
```

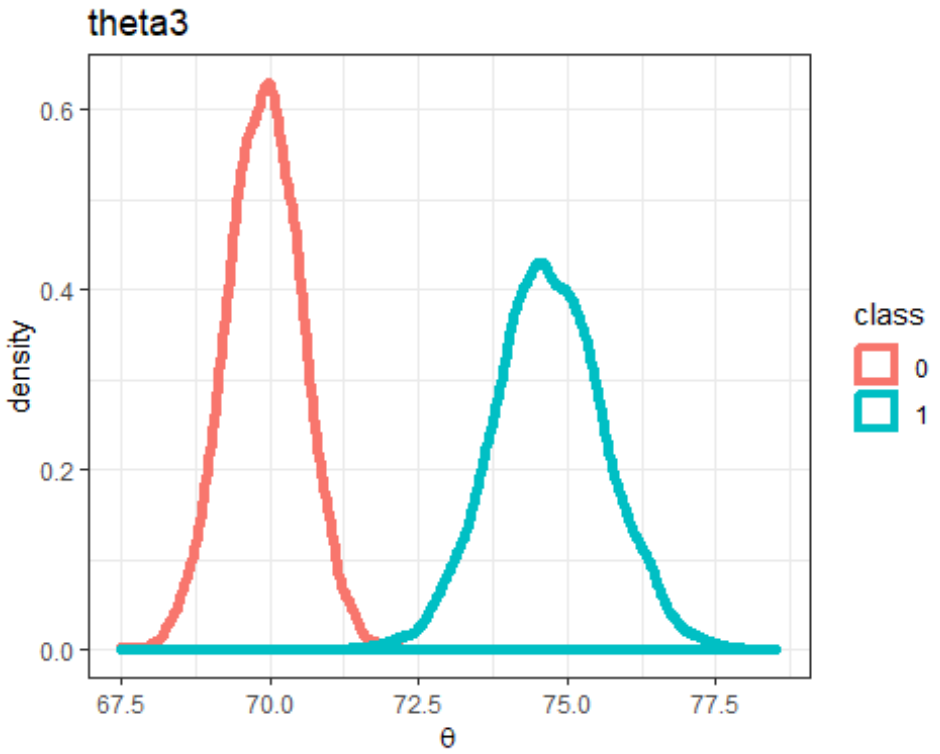
```
  for (j in 1:n_n) {
    s_theta_n=s_theta_n+(n[j,]-theta_n[i,])%*%t(n[j,]-theta_n[i,])
  }
  sn_n=s0_n+s_theta_n
  sigma_n[,,i]=solve(rwish(v=nun_n,S=solve(sn_n)))
}

theta.dt<-data.frame(theta_d,class=1)
names(theta.dt)=c(paste('theta',as.character(1:7),sep = ''),'class')
a<-cbind.data.frame(theta_n,0)
names(a)=c(paste('theta',as.character(1:7),sep = ''),'class')
theta.dt<-rbind.data.frame(theta.dt,a)
theta.dt$class<-as.factor(theta.dt$class)

library(ggplot2)
ggplot(theta.dt)+geom_density(aes(x=theta1,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta1')
```



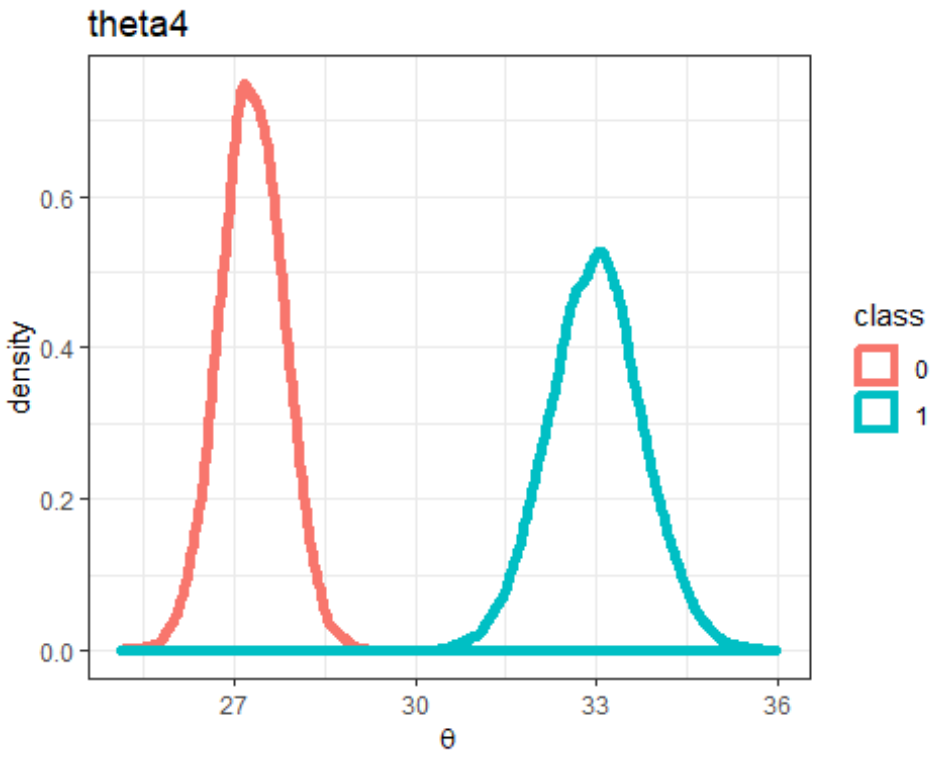

```
ggplot(theta.dt)+geom_density(aes(x=theta2,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta2')
```



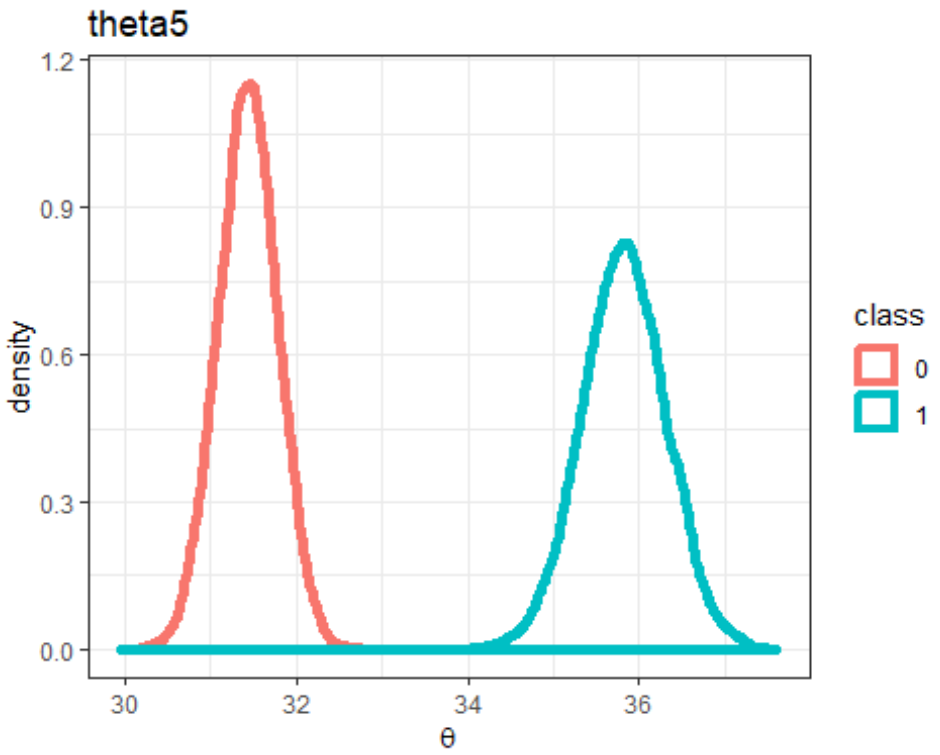

```
ggplot(theta.dt)+geom_density(aes(x=theta3,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta3')
```



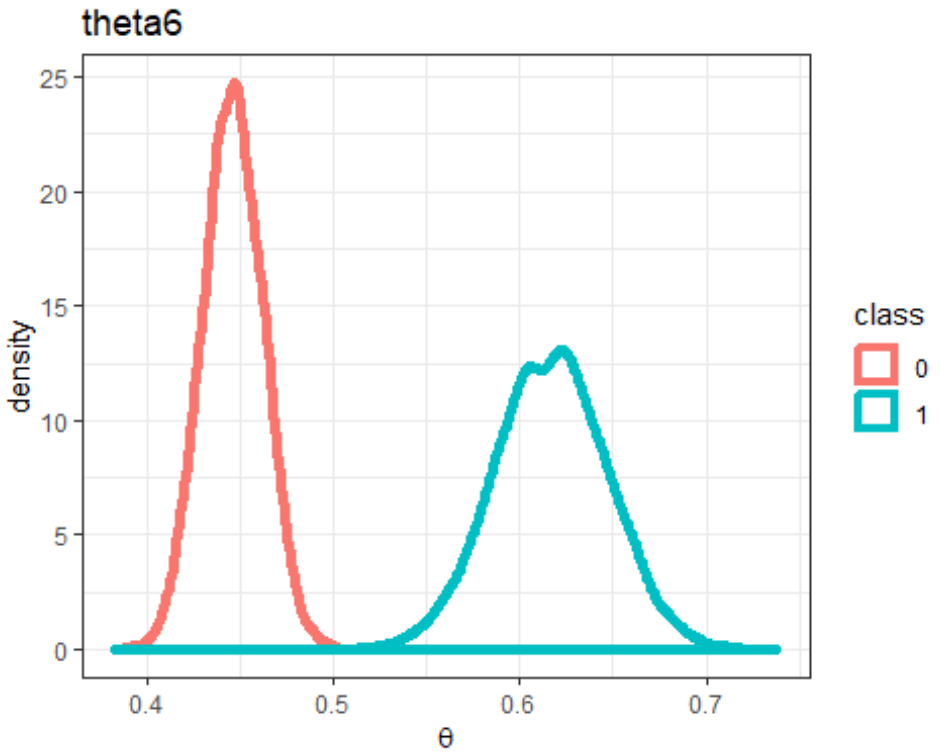
```
ggplot(theta.dt)+geom_density(aes(x=theta4,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta4')
```



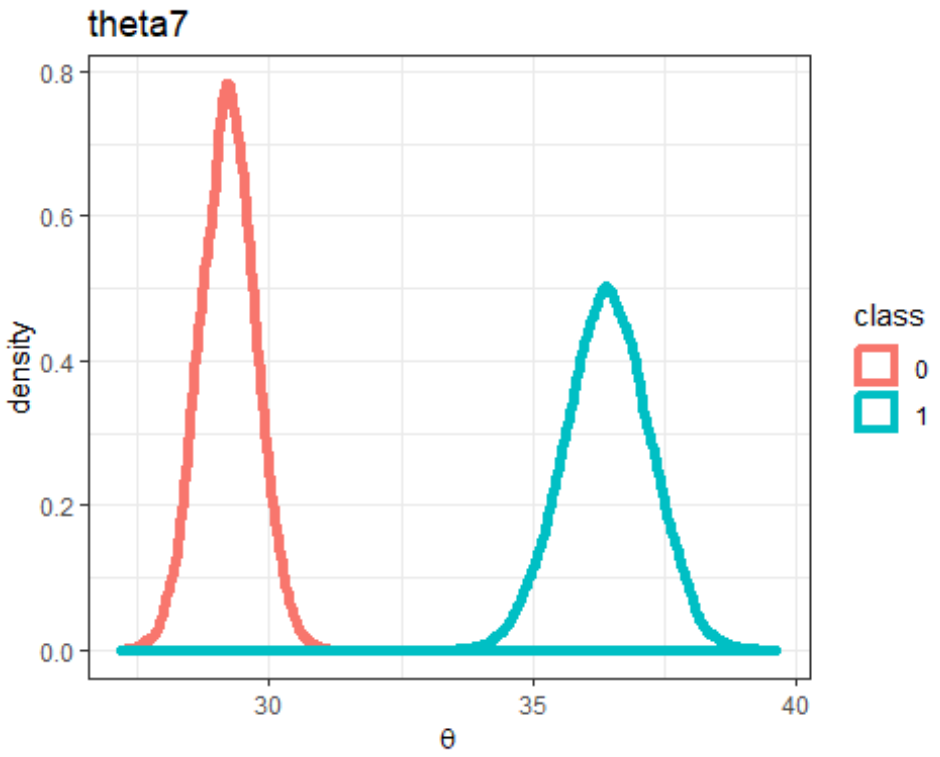

```
ggplot(theta.dt)+geom_density(aes(x=theta5,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta5')
```




```
ggplot(theta.dt)+geom_density(aes(x=theta6,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta6')
```



```
ggplot(theta.dt)+geom_density(aes(x=theta7,color=class),size=2)+theme_b
w()+xlab(expression(theta))+labs(title = 'theta7')
```



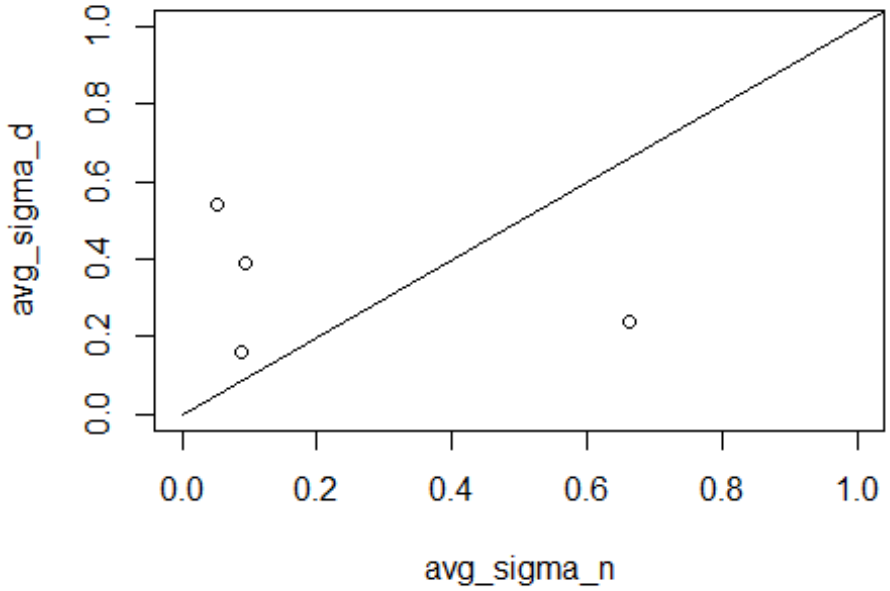

```
m<-c()
for (i in 1:7) {
  m[i]<-mean(theta.dt[theta.dt[,8]=='1',][,i]>theta.dt[theta.dt[,8]=='0
',][,i])
}
paste('p(theta_d,',1:7,'>theta_n,',1:7,') is ',m,sep='')%>%print()
```

```
## [1] "p(theta_d,1>theta_n,1) is 1" "p(theta_d,2>theta_n,2) is 1"
## [3] "p(theta_d,3>theta_n,3) is 1" "p(theta_d,4>theta_n,4) is 1"
## [5] "p(theta_d,5>theta_n,5) is 1" "p(theta_d,6>theta_n,6) is 1"
## [7] "p(theta_d,7>theta_n,7) is 1"
```

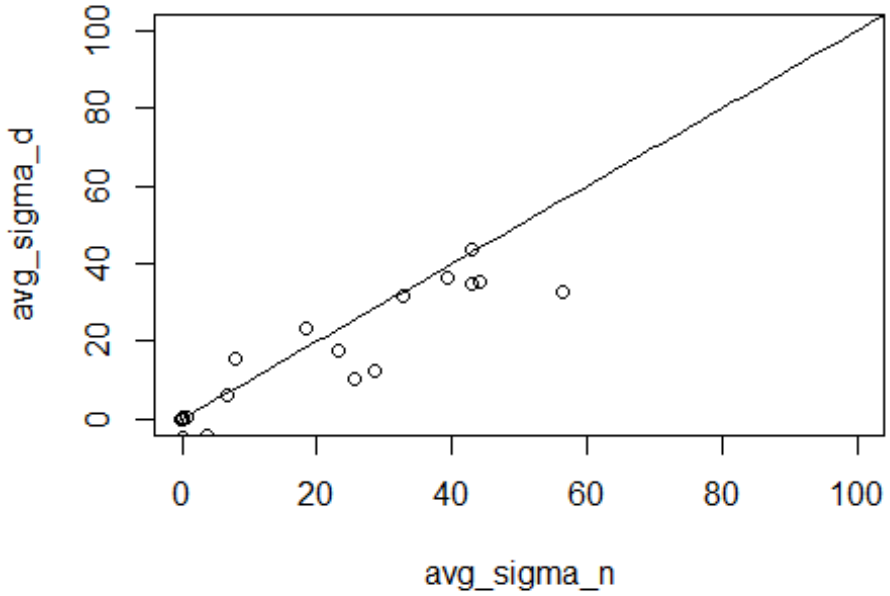
All the variables seem to be different between 2 groups.

2.(b).

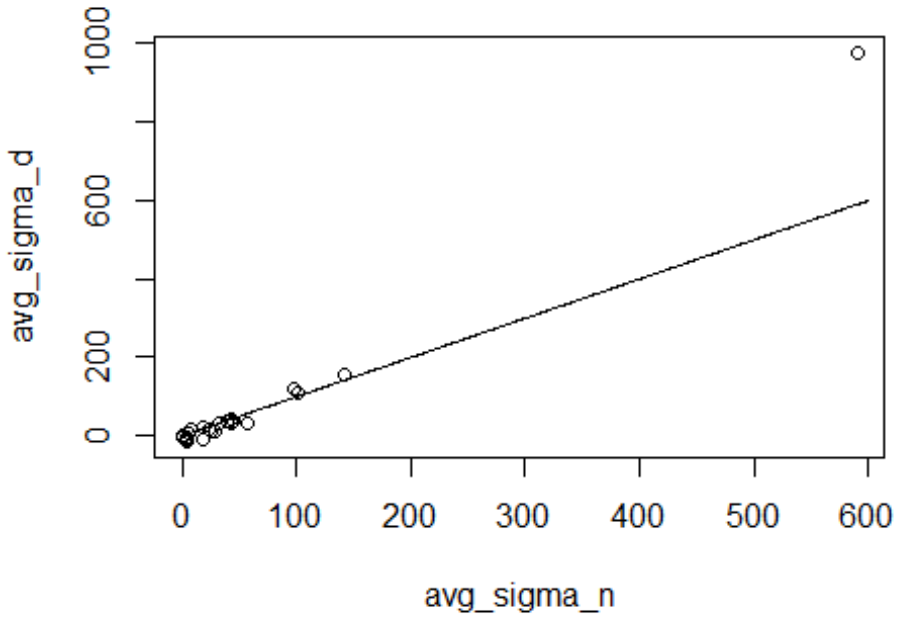
```
avg_sigma_d<-apply(sigma_d,c(1,2),mean)
avg_sigma_n<-apply(sigma_n,c(1,2),mean)
avg_sigma_d<-as.vector(avg_sigma_d)
avg_sigma_n<-as.vector(avg_sigma_n)

plot(avg_sigma_n,avg_sigma_d,xlim = c(0,1),ylim=c(0,1))
lines(0:600,0:600)
```




```
plot(avg_sigma_n,avg_sigma_d,xlim = c(0,100),ylim=c(0,100))
lines(0:600,0:600)
```



```r
plot(avg_sigma_n,avg_sigma_d)
lines(0:600,0:600)
```




```r
avg_sigma_d[9]#variance of glu for diabetes
```

```
## [1] 977.0127
```

```r
avg_sigma_n[9]#variance of glu for non-diabetes
```

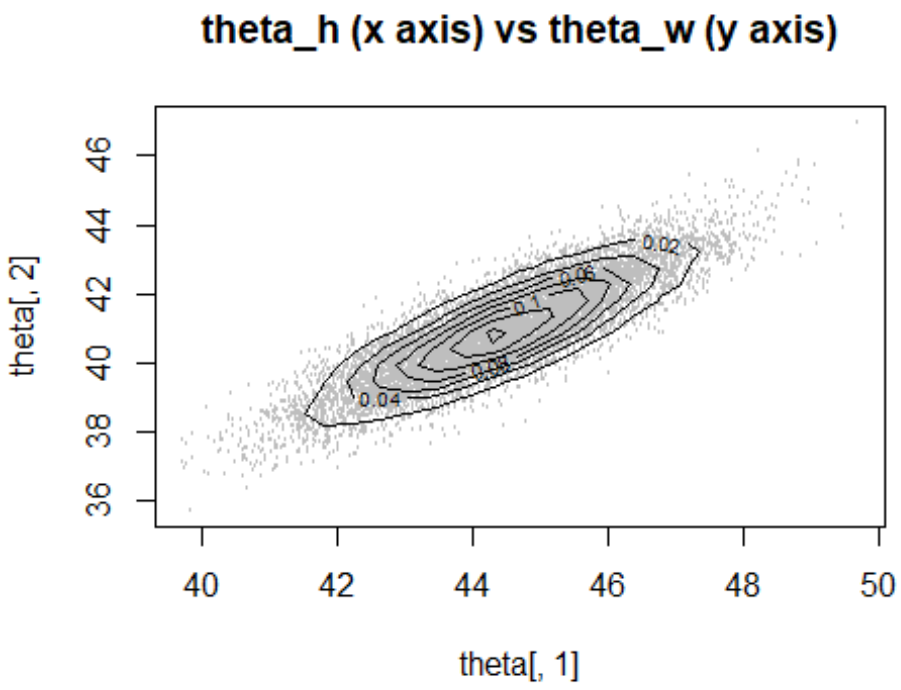
```
## [1] 590.3939
```

The line in the plot is line of y=x.
From the plot, we could find that most entries of sigma_d and sigma_n are almost same, except for the last point. The main difference is the variance of variable 'glu' measuring the percentage of glucose. For people with diabetes, the variance of glucose percentage is apparently higher than people without diabetes.

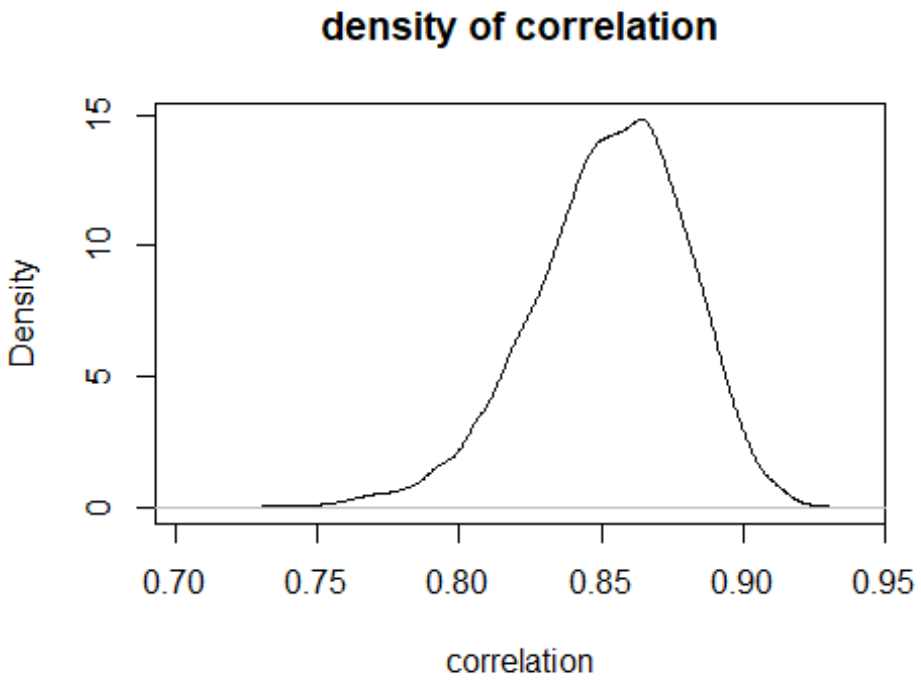
3.

```r
age<-read.table('./agehw.txt',header = T)
age<-as.matrix(age)
names(age)<-c('husband','wife')

#set start values
mu0=c(0,0)
lambda0=diag(2)*10^5
nu0=3
```

```r
s0=1000*diag(2)
n=dim(age)[1]
y.bar=apply(age,2,mean)
s=10000

theta=matrix(nrow=s,ncol=2)
sigma<-array(dim = c(dim(s0),s))
theta[1,]=y.bar[1:2]
sigma[,,1]=cov(age)


#start sampling
library(Rfast)
library(monomvn)

set.seed(123)
for (i in 2:s) {
  lambdan=solve(solve(lambda0)+n*solve(sigma[,,i-1]))
  lambdan[upper.tri(lambdan)]=0
  lambdan=lambdan+t(lambdan)
  diag(lambdan)=diag(lambdan)/2
  mun=lambdan%*%(solve(lambda0)%*%mu0+n*solve(sigma[,,i-1])%*%y.bar)
  theta[i,]=rmvnorm(n=1,mu=mun,sigma=lambdan)

  nun=nu0+n
  s_theta=0
  for (j in 1:n) {
    s_theta=s_theta+(age[j,]-theta[i,])%*%t(age[j,]-theta[i,])
  }
  sn=s0+s_theta
  sigma[,,i]=solve(rwish(v=nun,S=solve(sn)))
}
colnames(theta)<-names(age)[1:2]

library(LaplacesDemon)
joint.density.plot(theta[,1], theta[,2],Title = 'theta_h (x axis) vs th
eta_w (y axis)')
```




```r
corr<-c()
for (i in 1:s) {
  corr[i]<-cov2cor(sigma[,,i])[1,2]
}
plot(density(corr),main = 'density of correlation',xlab = 'correlation')
```




```
cat('The 95% CI for theta_h is','[',quantile(theta[,1],c(0.025,0.975)),
']\n')%>%print()
```

```
## The 95% CI for theta_h is [ 41.66887 47.22755 ]
## NULL
```

```
cat('The 95% CI for theta_w is','[',quantile(theta[,2],c(0.025,0.975)),
']\n')%>%print()
```

```
## The 95% CI for theta_w is [ 38.26553 43.54858 ]
## NULL
```

```
cat('The 95% CI for correlation is','[',quantile(corr,c(0.025,0.975)),']
')%>%print()
```

```
## The 95% CI for correlation is [ 0.7921709 0.8992922 ]NULL
```