Homework9_STA602

ElenaW.

10/31/2021

```
library(mvtnorm) # for drawing multivariate normal
library(MCMCpack) # for drawing inverse-Wishart
```

```
## Loading required package: coda

## Loading required package: MASS

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

## ## Copyright (C) 2003-2021 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)
## ##
```

```
library(ggplot2)
```

1. PH 7.5

a)

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

```
thetaA = mean(interexp$yA, na.rm = T)
thetaB = mean(interexp$yB, na.rm = T)
sigmaA = var(interexp$yA, na.rm = T)
sigmaB = var(interexp$yB, na.rm = T)
cor_AB = cor(na.omit(interexp))
```

b)

```
# impute B
i.miss_B = which(is.na(interexp$yB))
interexp$yB[i.miss_B] = thetaB + (interexp$yA[i.miss_B] - thetaA)*cor_AB[2,1]*sqrt(sigmaB/sigmaA)
```

```
# impute A
i.miss_A = which(is.na(interexp$yA))
interexp$yA[i.miss_A] = thetaA + (interexp$yB[i.miss_A] - thetaB)*cor_AB[2,1]*sqrt(sigmaA/sigmaB)
```

```
# paired sample t_test
t_test = t.test(interexp$yA, interexp$yB, paired = T, alternative = "two.sided")

t_test$p.value
```

```
## [1] 0.001769777
```

```
t_test$conf.int
```

```
## [1] -0.9850730 -0.2383347
## attr(,"conf.level")
## [1] 0.95
```

From the results above, we could see that the p-value is t-test is just 0.0018, which is really small, so we have 95% confidence to reject NULL and conclude that there is a difference between true mean of A and B (theta). The 95% confidence interval for the true difference of thetaA and thetaB is from -0.9850730 to -0.2383347.

c)

Using Jeffrey's prior and the function obtained from HW8

```
Gibbs_jeffrey = function(y.original){
n = nrow(y.original) # sample size
p = ncol(y.original) # dimensionality
I <- !is.na(y.original) # missingness indicator, TRUE if present, 0 if missing

# prior for theta
mu.0 = colMeans(y.original)
lambda.0 = cov(y.original)
# prior for sigma
nu.0 = p + 2
S0 = cov(y.original)

# Gibbs sampling
niter = 10000 # total number of iteration
nburnin = 1000 # 1000 burn-in step

ybar.original <- apply(y.original,2,mean,na.rm=TRUE) # the column means of the original data
y <- y.original ## y holds the imputed data (y.obs,y.mis)
for (i in 1:p) {
y[I[,i]==0,i] <- ybar.original[i]
}

## Proceed as before like there are no missing data
ybar <- apply(y,2,mean)
nu.n <- 1 + n

THETA = matrix(NA, nrow = niter, ncol = p) # matrix for storing the draws for theta
colnames(THETA) = c("theta1", "theta2")
THETA.init = ybar # Initial values set to sample mean
THETA.curr = THETA.init # the theta value at current iteration
SIGMA = matrix(NA, nrow = niter, ncol = p*p) # matrix for storing the draws for sigma
colnames(SIGMA) = c("sigma1", "sigma2", "sigma21", "sigma12")
SIGMA.int = cov(y) # initial value set to sample covariance
SIGMA.curr = SIGMA.int # the sigma value at current iteration
### Start Gibbs sampling
for (t in 1:niter){
## Update theta
lambda.n = SIGMA.curr/n
mu.n = ybar
THETA.curr = rmvnorm(1, mean = mu.n, sigma = lambda.n) # random multivariate normal
## Update sigma
S.theta = (t(y)-c(THETA.curr)) %*% t(t(y)-c(THETA.curr))
SIGMA.curr <- riwish(v=nu.n,S=S.theta)

## Impute the missing data
for (i in 1:n) {
var.obs = which(I[i,]) ## which variables are observed
var.mis = which(!I[i,]) ## which variables are missing
if (length(var.mis) > 0){ ## if there are missing values
SIGMA.obs <- SIGMA.curr[var.obs,var.obs] # Sigma11
SIGMA.mis <- SIGMA.curr[var.mis,var.mis] # Sigma22
SIGMA.mis.obs <- SIGMA.curr[var.mis,var.obs] # Sigma21
SIGMA.obs.mis <- t(SIGMA.mis.obs) # Sigma12
y[i,var.mis] <- rnorm(1, mean=THETA.curr[var.mis]+
SIGMA.mis.obs%*%solve(SIGMA.obs,y[i,var.obs]-THETA.curr[var.obs]),
sd=sqrt(SIGMA.mis-SIGMA.mis.obs%*%solve(SIGMA.obs,SIGMA.obs.mis)))
}
}
ybar <- apply(y,2,mean)

## Save the current iteration
THETA[t,] <- THETA.curr
SIGMA[t,] <- SIGMA.curr

}
return(list(theta = THETA,sigma = SIGMA))
}
```

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

```
theta = data.frame(sim_Jeffrey$theta)
diff_theta = theta$theta1 - theta$theta2
```

```
mean(diff_theta)
```

```
## [1] -0.6134971
```

```
mean(theta$theta1>theta$theta2)
```

```
## [1] 0.0371
```

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

```
##       2.5%      97.5%
## -1.3031237  0.0625644
```

From the results above, the mean of the difference of thetas is -0.617, and the probability of theta1 greater than theta2 is really small, which means that thetas generated from Gibbs sampling are different and it's really possible that the true theta A is smaller than theta B. The confidence interval generated from Gibbs sampling is slightly larger than b), which is from -1.30 to 0.061.

2. PH 7.6

a)

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

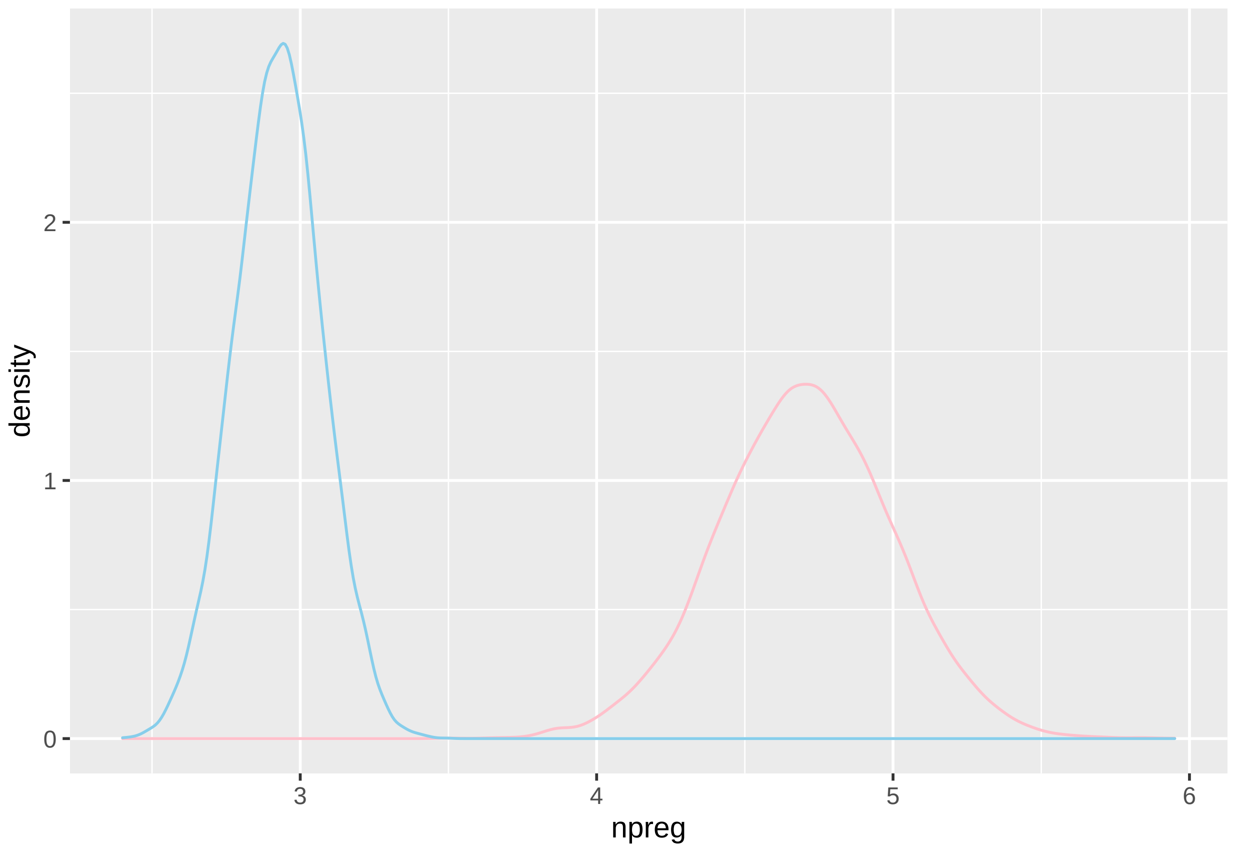
```
# separate groups and delete the last column
dia = diabetes[which(diabetes$diabetes == "Yes"),]
nodia = diabetes[which(diabetes$diabetes == "No"),]
dia$diabetes = NULL
nodia$diabetes = NULL
```

```
set.seed(0)
Gibbs = function(y){
n = nrow(y) # sample size
p = ncol(y) # dimensionality
# prior for theta
mu.0 = colMeans(y)
lambda.0 = cov(y)
# prior for sigma
nu.0 = 9
S0 = cov(y)
ybar = colMeans(y)
nu.n = nu.0 + n
# Gibbs sampling
niter = 10000 # total number of iteration
nburnin = 1000 # 1000 burn-in step
THETA = matrix(NA, nrow = niter, ncol = p) # matrix for storing the draws for theta
colnames(THETA) = c("theta1", "theta2","theta3","theta4", "theta5", "theta6","theta7")
THETA.init = ybar # Initial values set to sample mean
THETA.curr = THETA.init # the theta value at current iteration
SIGMA = matrix(NA, nrow = niter, ncol = p*p) # matrix for storing the draws for sigma
#colnames(SIGMA) = c("sigma1", "sigma2", "sigma21", "sigma12")
SIGMA.int = cov(y) # initial value set to sample covariance
SIGMA.curr = SIGMA.int # the sigma value at current iteration
### Start Gibbs sampling
for (t in 1:niter){
## Update theta
lambda.n = solve((n*solve(SIGMA.curr))+solve(lambda.0))
mu.n = lambda.n %*% (n*solve(SIGMA.curr,ybar)+solve(lambda.0,mu.0))
THETA.curr = rmvnorm(1, mean = mu.n, sigma = lambda.n) # random multivariate normal
## Update sigma
S.theta = (t(y)-c(THETA.curr)) %*% t(t(y)-c(THETA.curr))
SIGMA.curr <- riwish(v=nu.n,S=S0+S.theta)
## Save the current iteration
THETA[t,] <- THETA.curr
SIGMA[t,] <- SIGMA.curr
}
return(list(theta = THETA,sigma = SIGMA))
}
```

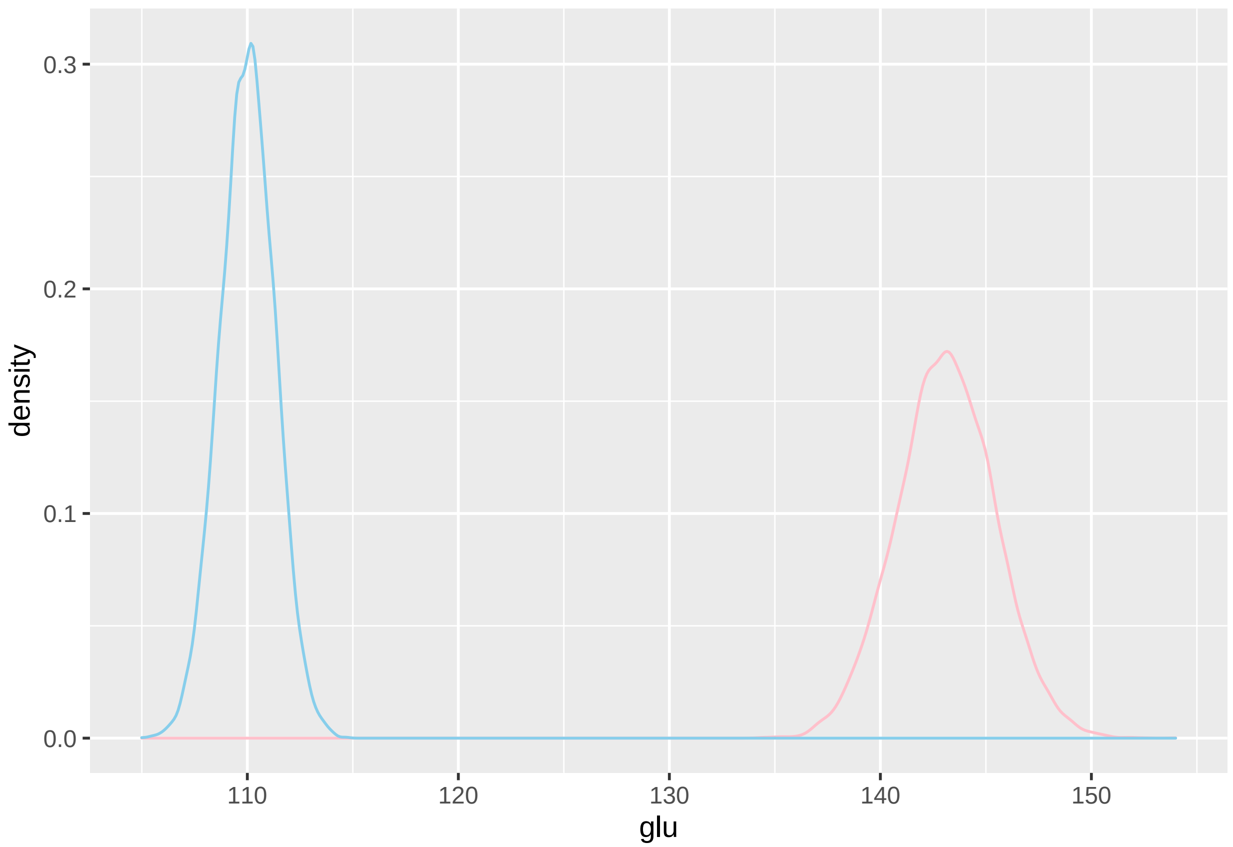
```
sim_dia = Gibbs(dia)
sim_nodia = Gibbs(nodia)
```

Compare the posterior distribution between d,j and n,j, pink is for diabetes, and blue is for non-diabetes.

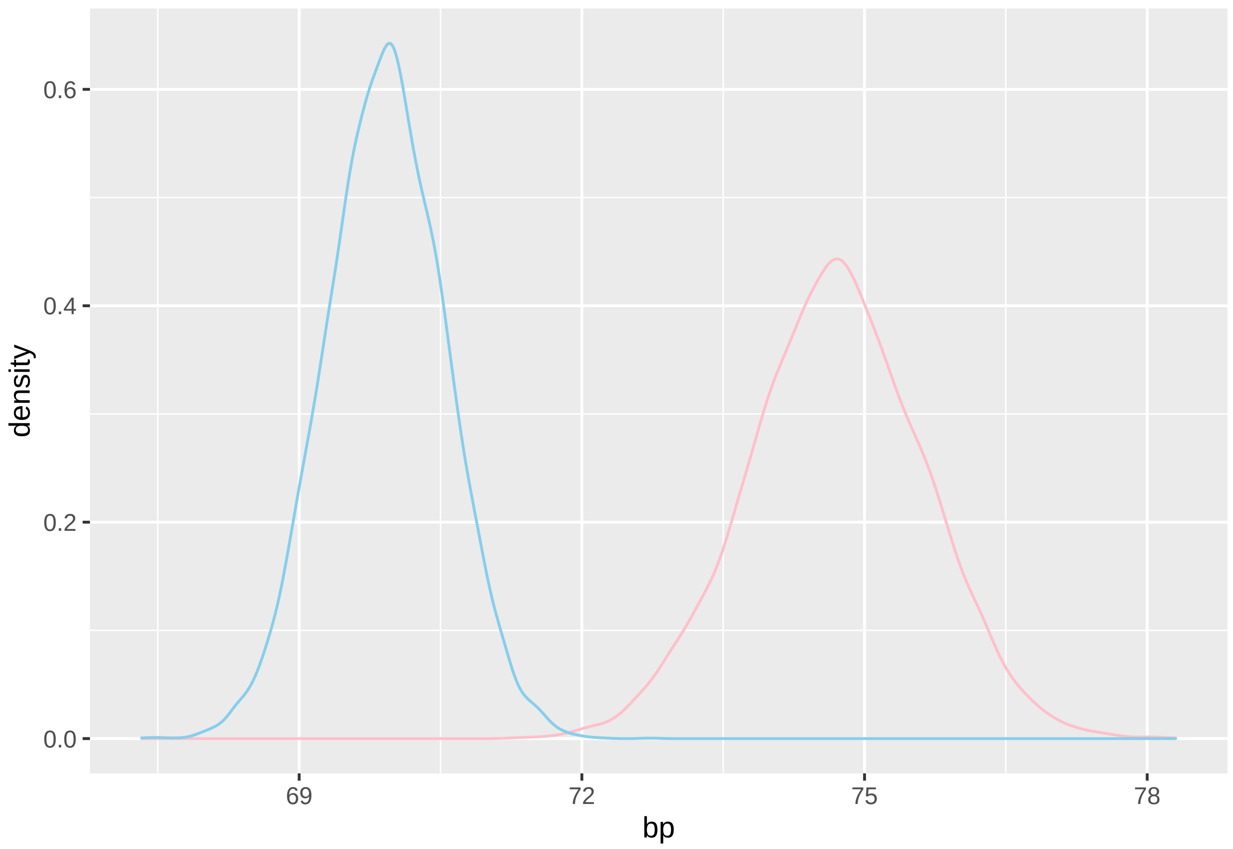
```
par(mfrow = c(1,7))
theta_dia = data.frame(sim_dia$theta)
theta_nodia = data.frame(sim_nodia$theta)
ggplot()+
  geom_density(aes(x = theta_dia$theta1), color = "pink")+
  geom_density(aes(x = theta_nodia$theta1),color = "skyblue")+
  labs(x = "npreg")
```



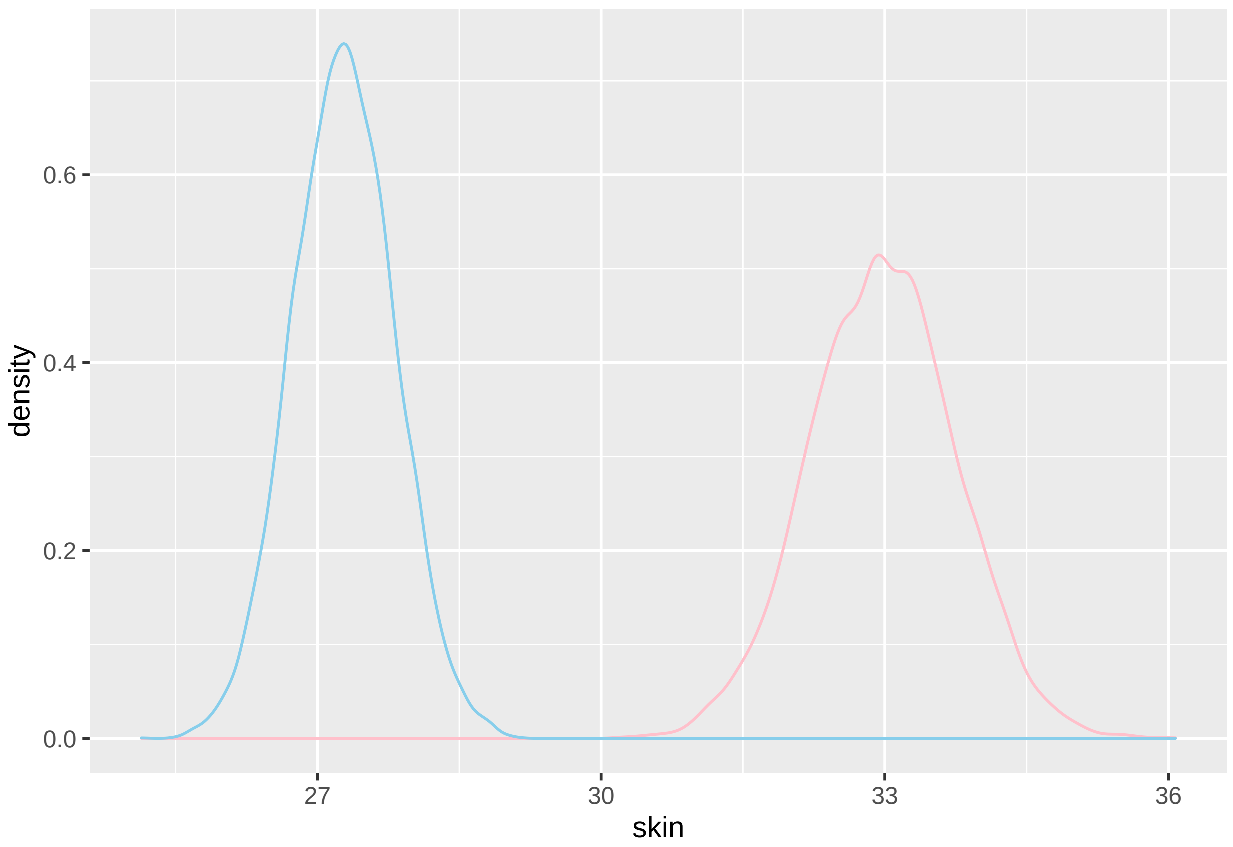
```
ggplot()+
  geom_density(aes(x = theta_dia$theta2), color = "pink")+
  geom_density(aes(x = theta_nodia$theta2),color = "skyblue")+
  labs(x = "glu")
```



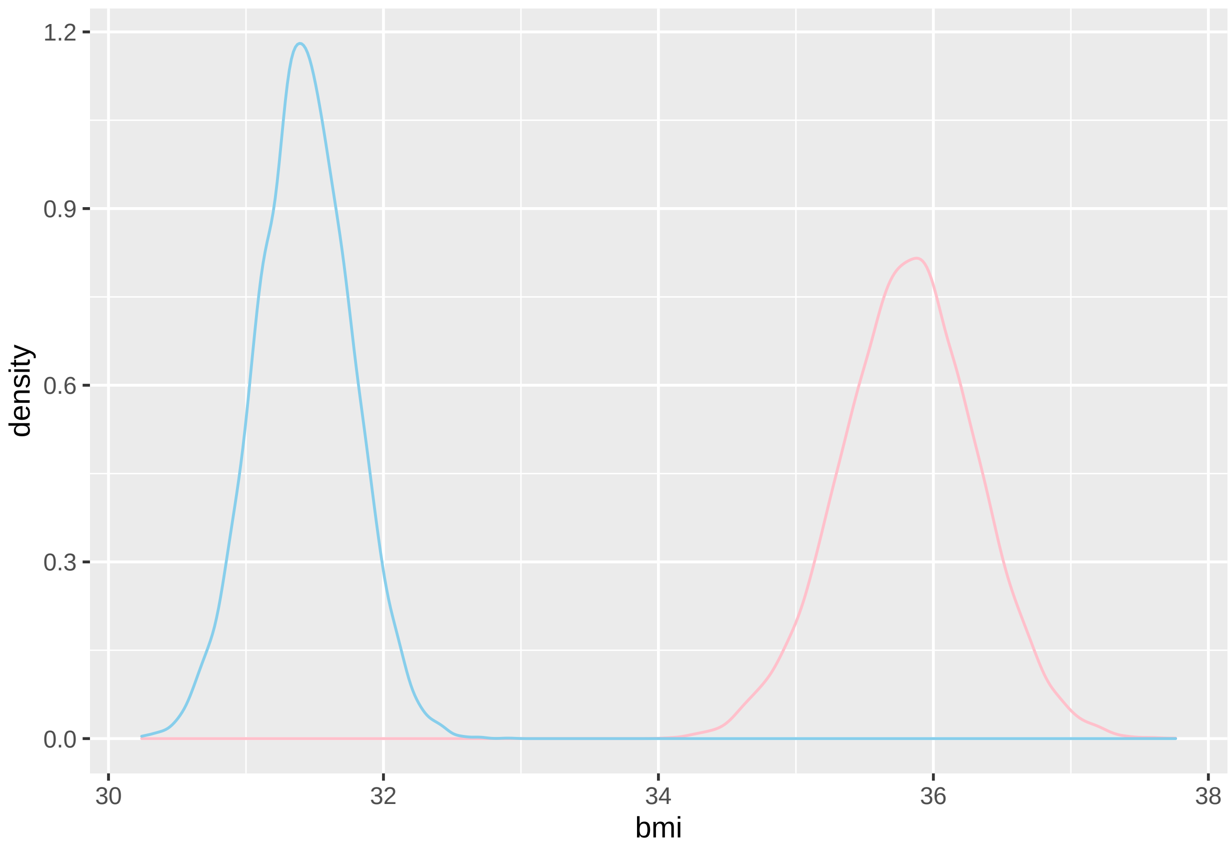
```
ggplot()+
  geom_density(aes(x = theta_dia$theta3), color = "pink")+
  geom_density(aes(x = theta_nodia$theta3),color = "skyblue")+
  labs(x = "bp")
```



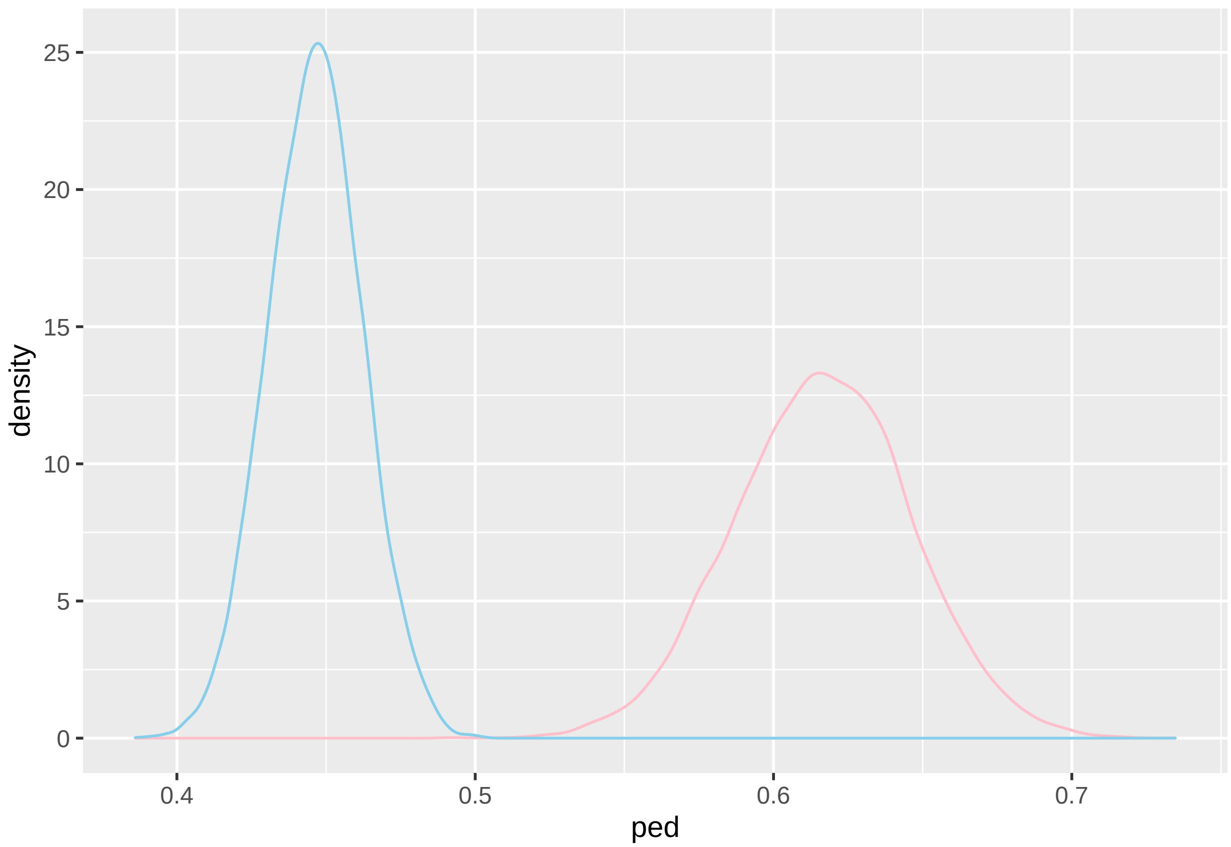
```
ggplot()+
  geom_density(aes(x = theta_dia$theta4), color = "pink")+
  geom_density(aes(x = theta_nodia$theta4),color = "skyblue")+
  labs(x = "skin")
```



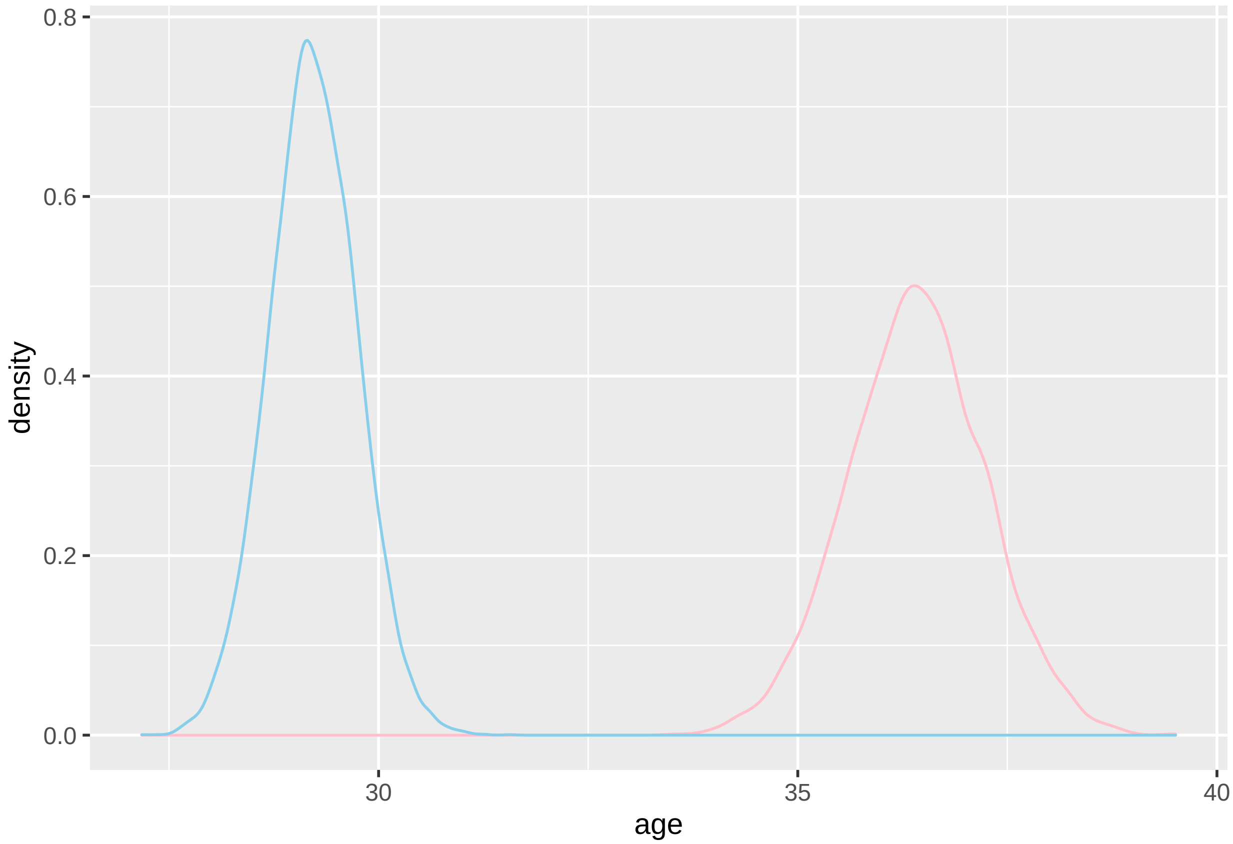
```
ggplot()+
  geom_density(aes(x = theta_dia$theta5), color = "pink")+
  geom_density(aes(x = theta_nodia$theta5),color = "skyblue")+
  labs(x = "bmi")
```



```
ggplot()+
  geom_density(aes(x = theta_dia$theta6), color = "pink")+
  geom_density(aes(x = theta_nodia$theta6),color = "skyblue")+
  labs(x = "ped")
```



```
ggplot()+
  geom_density(aes(x = theta_dia$theta7), color = "pink")+
  geom_density(aes(x = theta_nodia$theta7),color = "skyblue")+
  labs(x = "age")
```



From the graphs above, all of them separates well, but the most difference variable between two groups is bp since the overlap is the least and the smallest difference variable is glue since the overlap is the most.

For Pr(d,j > n,j |Y), from the results below, we could see that theta for diabetes is larger than non-diabetes in all variables, which means that the probability of that is 1 in all variables.

```
for (i in 1:7){
  print(mean(theta_dia[,i]>mean(theta_nodia[,i])))
}
```

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

b)

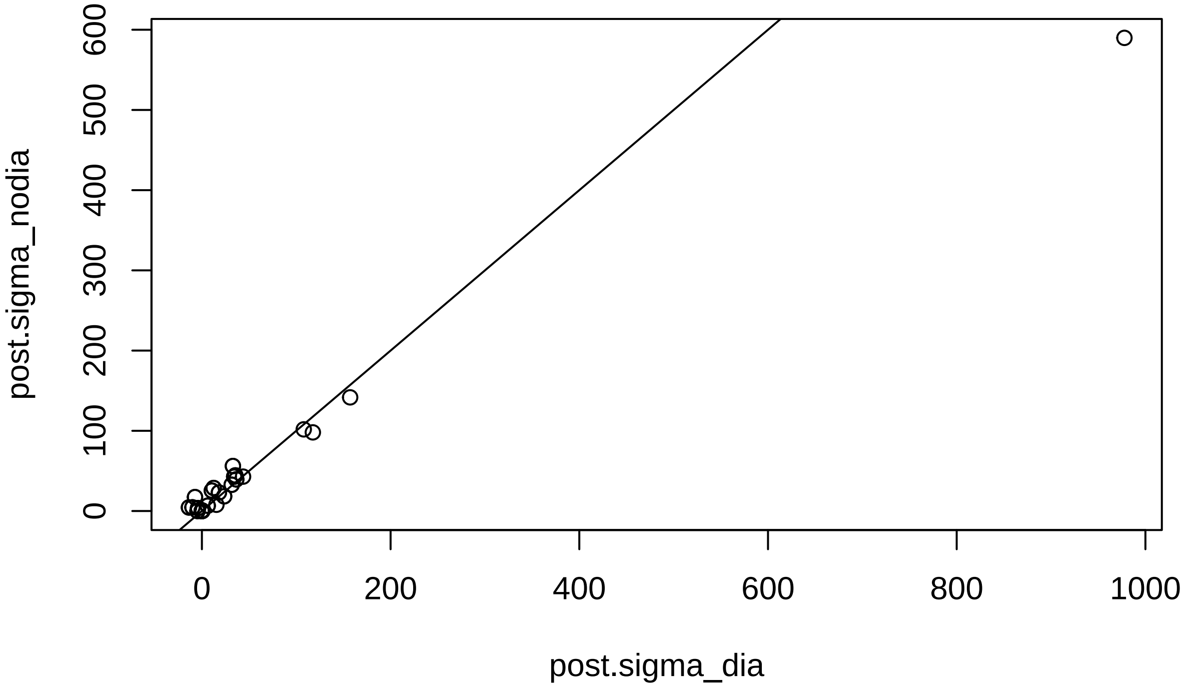
```
sigma_dia = data.frame(sim_dia$sigma)
sigma_nodia = data.frame(sim_nodia$sigma)
post.sigma_dia = as.numeric(apply(sigma_dia[,1:49],2,mean))
post.sigma_nodia = as.numeric(apply(sigma_nodia[,1:49],2,mean))
post.sigma_dia
```

```
##  [1]  15.36492100 -10.01328407   6.18613861  -4.10480301  -4.66247411
##  [6]  -0.09544827  23.57548301 -10.01328407 977.76180443  32.87062580
## [11]  31.61661486  10.56722495   0.25162059  34.24286448   6.18613861
## [16]  32.87062580 157.13843163  12.46287413  18.04317920  -0.17153028
## [21]  36.42325369  -4.10480301  31.61661486  12.46287413 107.94894271
## [26]  35.44157695   0.53852324  -7.31402371  -4.66247411  10.56722495
## [31]  18.04317920  35.44157695  43.63514593   0.39003338 -13.74265417
## [36]  -0.09544827   0.25162059  -0.17153028   0.53852324   0.39003338
## [41]   0.15917592  -0.15183001  23.57548301  34.24286448  36.42325369
## [46]  -7.31402371 -13.74265417  -0.15183001 117.60113728
```

```
post.sigma_nodia
```

```
##  [1]   7.774399827   4.578965362   6.632399621   3.627650312  -0.005812384
##  [6]  -0.041503082  18.308377615   4.578965362 589.847811099  56.066109893
## [11]  32.720516556  25.449448053   0.666378401  42.841664579   6.632399621
## [16]  56.066109893 141.694045129  28.667337966  23.054106524  -0.131890666
## [21]  39.422270188   3.627650312  32.720516556  28.667337966 101.783068467
## [26]  44.365812959   0.055936427  17.351335666  -0.005812384  25.449448053
## [31]  23.054106524  44.365812959  42.865770704   0.095261549   4.421372284
## [36]  -0.041503082   0.666378401  -0.131890666   0.055936427   0.095261549
## [41]   0.089354857   0.065032112  18.308377615  42.841664579  39.422270188
## [46]  17.351335666   4.421372284   0.065032112  98.039407631
```

```
plot(x = post.sigma_dia,y = post.sigma_nodia)+
  abline(coef = c(0,1))
```



```
## integer(0)
```

From the plot above, we could see that almost all the point is around the 45-degree line, which means that there are not many overall differences for posterior covariance.