HW6 Solution

Jitong

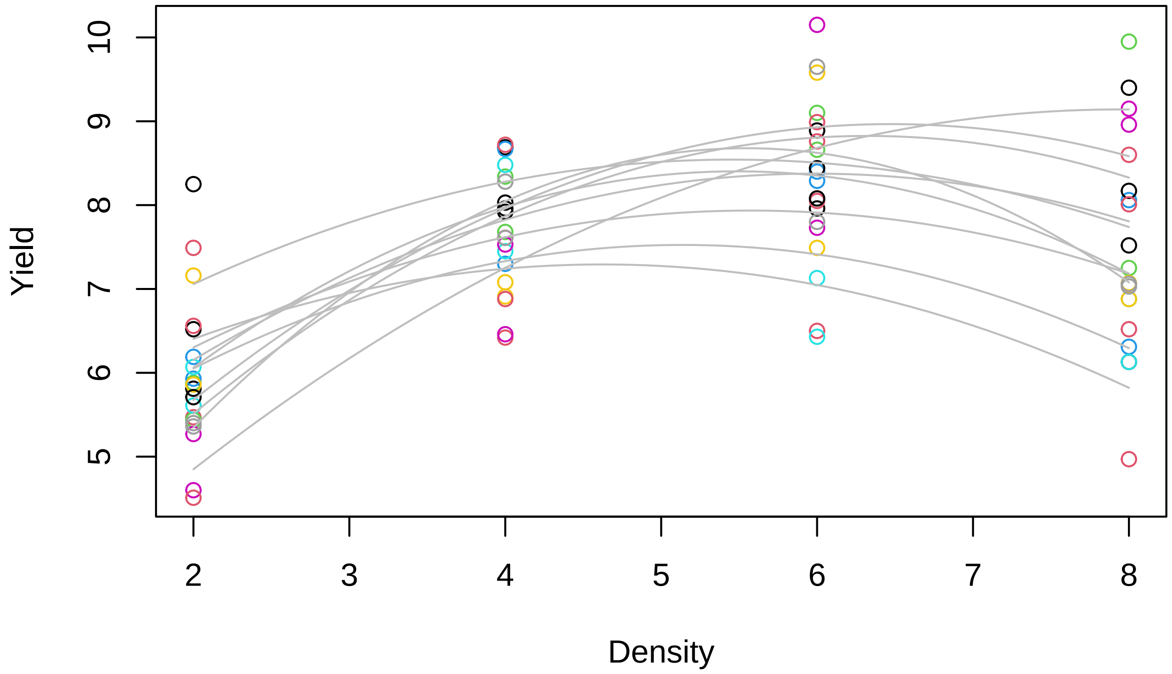
2023-04-25

Question 1

a)

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

```
ids<-sort(unique(data_p1$plot))
m<-length(ids)
Y<-list() ; X<-list() ; N<-NULL
for(j in 1:m)
{
  Y[[j]]<-data_p1[data_p1$plot==ids[j], 3]
  N[j]<- sum(data_p1$plot==ids[j])
  xj<-data_p1[data_p1$plot==ids[j], 2]
  X[[j]]<-cbind(rep(1,N[j]), xj, xj^2)
}
GRID = data.frame(cbind(1, seq(2,8,0.1), seq(2,8,0.1)^2))
x_grid =seq(2,8,0.1)
#### OLS fits
FIT.LINE = NULL
S2.LS<-BETA.LS<-NULL
for(j in 1:m) {
  fit<-lm(Y[[j]]~-1+X[[j]])
  BETA.LS<-rbind(BETA.LS,c(fit$coef))
  S2.LS<-c(S2.LS, summary(fit)$sigma^2)
}
PRED = BETA.LS %*% t(GRID)
plot(data_p1[,2], data_p1[,3], col = data_p1[,1], xlab = 'Density', ylab = 'Yield')
for(j in 1:m){
  lines(x_grid, PRED[j,], type ='l', col = 'gray')
}
```



```
apply(BETA.LS,2,mean)
```

```
##   X[[j]] X[[j]]xj   X[[j]]
##  2.86875  1.85485 -0.15925
```

```
mean(S2.LS)
```

```
## [1] 0.7878099
```

```
cov(BETA.LS)
```

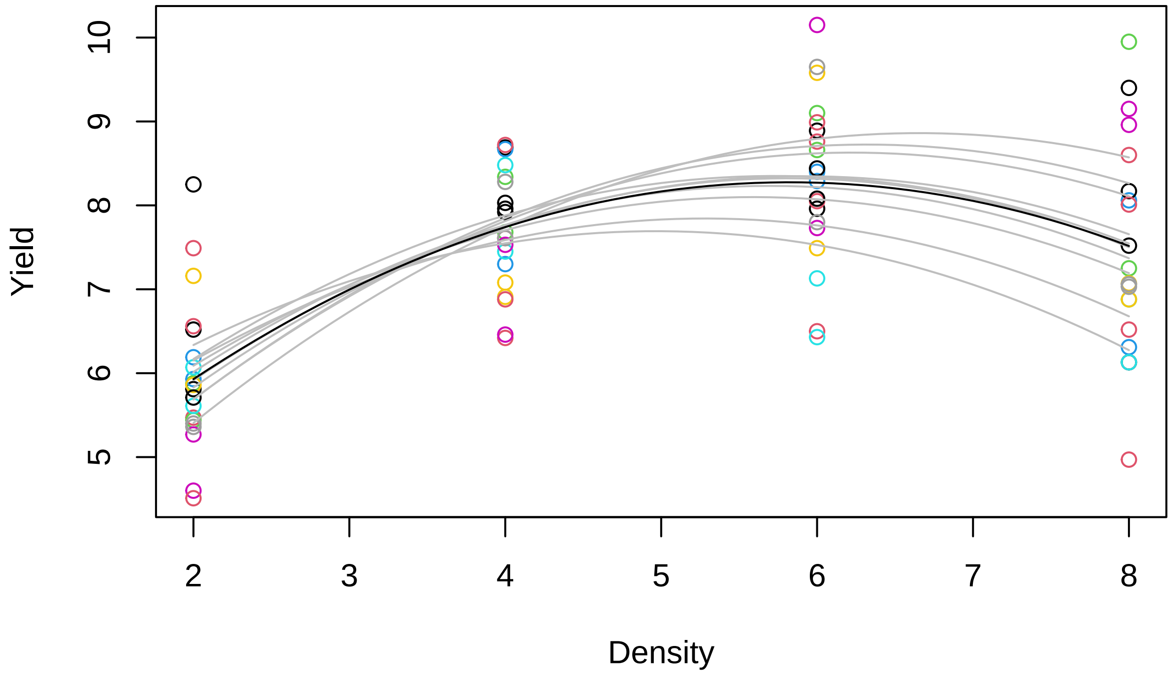
```
##               X[[j]]    X[[j]]xj       X[[j]]
## X[[j]]     2.00120764 -0.69321312  0.044309549
## X[[j]]xj -0.69321312  0.27555421 -0.020742679
## X[[j]]     0.04430955 -0.02074268  0.001968451
```

b, c)

```
## mvnormal simulation
rmvnorm<-function(n,mu,Sigma)
{
  E<-matrix(rnorm(n*length(mu)),n,length(mu))
  t( t(E%*%chol(Sigma)) +c(mu))
}
## Wishart simulation
rwish<-function(n,nu0,S0)
{
  sS0 <- chol(S0)
  S<-array( dim=c( dim(S0),n ) )
  for(i in 1:n)
  {
    Z <- matrix(rnorm(nu0 * dim(S0)[1]), nu0, dim(S0)[1]) %*% sS0
    S[,,i]<- t(Z)%*%Z
  }
  S[,,1:n]
}
## Setup
p<-dim(X[[1]])[2]
theta<-mu0<-apply(BETA.LS,2,mean)
nu0<-2 ; s2<-s20<-mean(S2.LS)
eta0<-4 ; Sigma<-S0<-L0<-cov(BETA.LS) ; BETA<-BETA.LS
THETA.b<-S2.b<-NULL
iL0<-solve(L0) ; iSigma<-solve(Sigma)
Sigma.ps<-matrix(0,p,p)
SIGMA.PS<-NULL
BETA.ps<-BETA*0
BETA.pp<-NULL
set.seed(1)
## MCMC
for(s in 1:10000) {
  ##update beta_j
  for(j in 1:m)
  {
    Vj<-solve( iSigma + t(X[[j]])%*%X[[j]]/s2 )
    Ej<-Vj%*%( iSigma%*%theta + t(X[[j]])%*%Y[[j]]/s2 )
    BETA[j,]<-rmvnorm(1,Ej,Vj)
  }
  ##
  ##update theta
  Lm<- solve( iL0 + m*iSigma )
  mum<- Lm%*%( iL0%*%mu0 + iSigma%*%apply(BETA,2,sum))
  theta<-t(rmvnorm(1,mum,Lm))
  ##
  ##update Sigma
  mtheta<-matrix(theta,m,p,byrow=TRUE)
  iSigma<-rwish(1, eta0+m, solve( S0+t(BETA-mtheta)%*%(BETA-mtheta) ) )
  ##
  ##update s2
  RSS<-0
  for(j in 1:m) { RSS<-RSS+sum( (Y[[j]]-X[[j]]%*%BETA[j,] )^2 ) }
  s2<-1/rgamma(1,(nu0+sum(N))/2, (nu0*s20+RSS)/2 )
  ##
  ##store results
  if(s%%10==0)
  {
    # cat(s,s2,"\n")
    S2.b<-c(S2.b,s2);THETA.b<-rbind(THETA.b,t(theta))
    Sigma.ps<-Sigma.ps+solve(iSigma) ; BETA.ps<-BETA.ps+BETA
    SIGMA.PS<-rbind(SIGMA.PS,c(solve(iSigma)))
    BETA.pp<-rbind(BETA.pp,rmvnorm(1,theta,solve(iSigma)) )
  }
```

```
}
```

```
BETA.PM <- BETA.ps/1000
BETA.PM.mean <- apply(BETA.PM, 2, mean)
PRED.bayesian = BETA.PM %*% t(GRID)
pred.bayesian.mean = BETA.PM.mean %*% t(GRID)
plot(data_p1[,2], data_p1[,3], col = data_p1[,1],xlab = 'Density', ylab = 'Yield')
for(j in 1:m){
  lines(x_grid, PRED.bayesian[j,], type ='l', col = 'gray')
}
lines(x_grid, pred.bayesian.mean, type ='l', col = 'black')
```

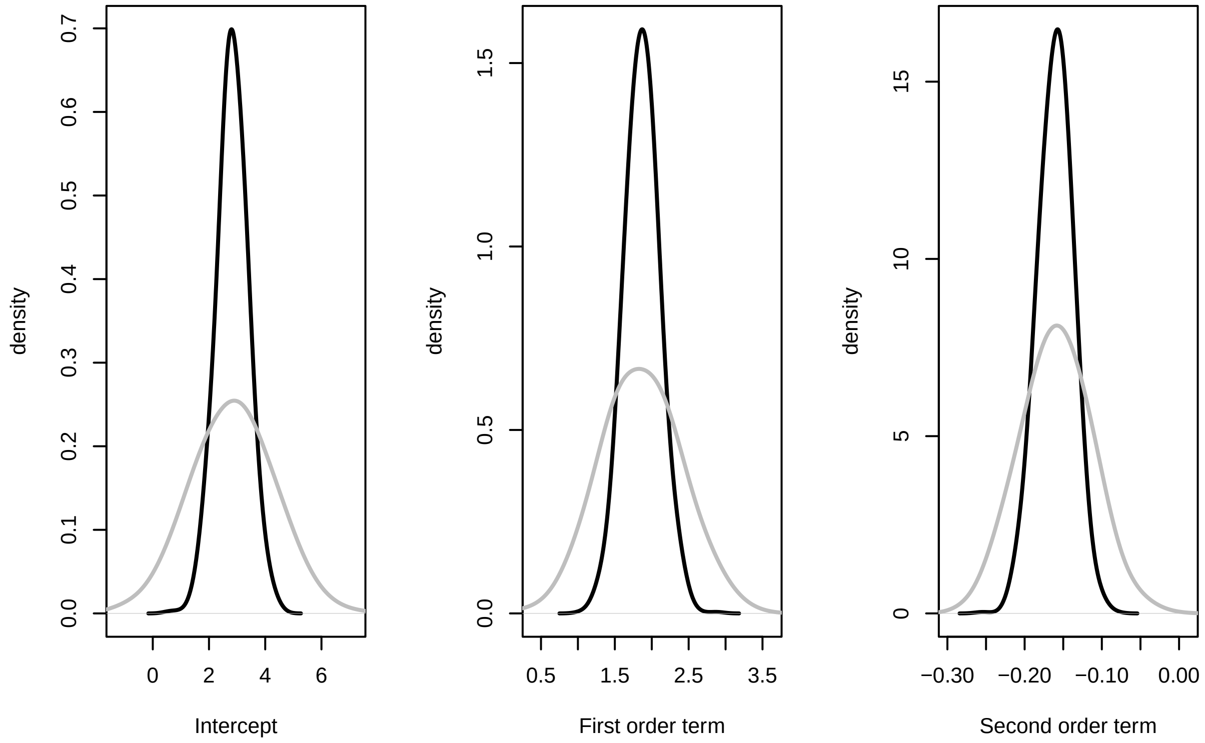


The fitted functions for different groups are closer to the overall fitted function.

d)

```
prior_theta = rmvnorm(1000, mu0, Sigma)
par(mfrow=c(1,3))
plot(density(THETA.b[,1],adj=2),xlim=range(BETA.pp[,1]),
main="",xlab="Intercept",ylab="density",lwd=2)
lines(density(prior_theta[,1],adj=2),col="gray",lwd=2)
plot(density(THETA.b[,2],adj=2),xlim=range(BETA.pp[,2]),
main="",xlab="First order term",ylab="density",lwd=2)
lines(density(prior_theta[,2],adj=2),col="gray",lwd=2)
plot(density(THETA.b[,3],adj=2),xlim=range(BETA.pp[,3]),
main="",xlab="Second order term",ylab="density",lwd=2)
```

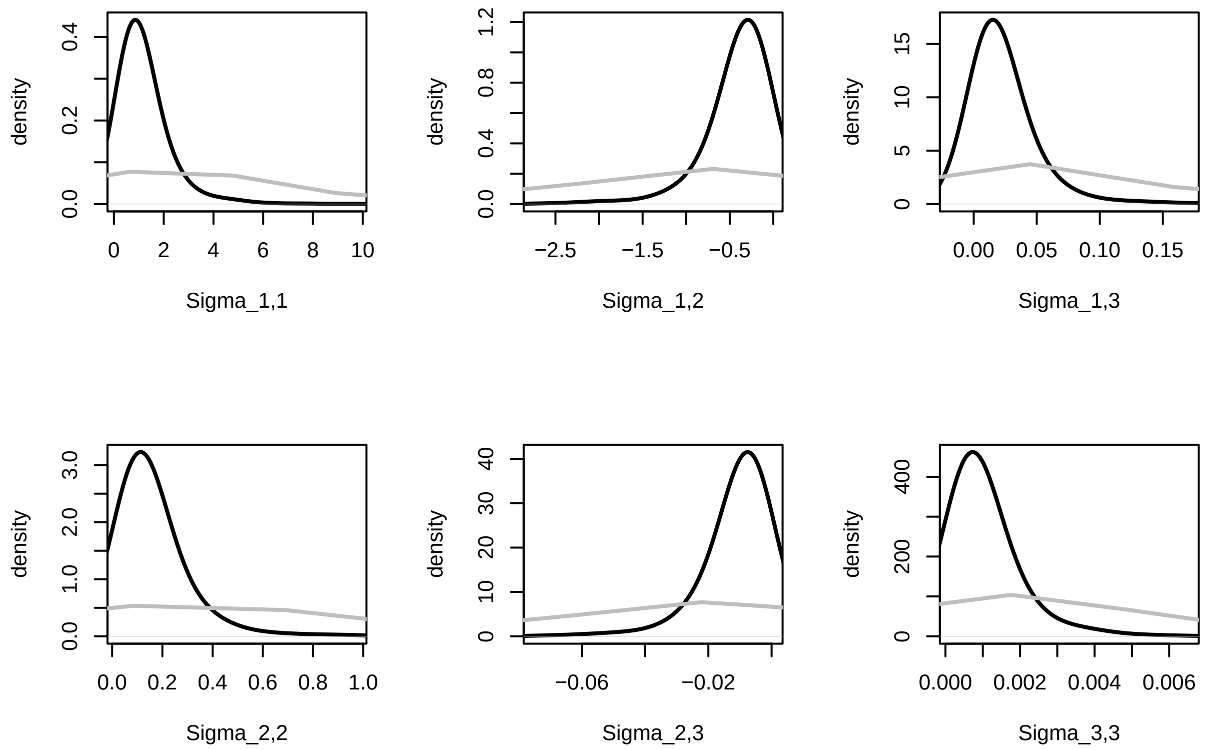
```
lines(density(prior_theta[,3],adj=2),col="gray",lwd=2)
```



```
SIGMA.prior.Sample = NULL
for(i in 1:1000){
  tmp = rwish(1, 4, solve(Sigma))
  SIGMA.prior.Sample<-rbind(SIGMA.prior.Sample,c(solve(tmp)))
}

par(mfrow=c(2,3))
plot(density(SIGMA.PS[,1],adj=5),xlim=range(SIGMA.PS[,1]),
main="",xlab="Sigma_1,1",ylab="density",lwd=2, col = 'black')
lines(density(SIGMA.prior.Sample[,1],adj=5),lwd=2, col = 'gray')
plot(density(SIGMA.PS[,2],adj=5),xlim=range(SIGMA.PS[,2]),
main="",xlab="Sigma_1,2",ylab="density",lwd=2, col = 'black')
lines(density(SIGMA.prior.Sample[,2],adj=5),lwd=2, col = 'gray')
plot(density(SIGMA.PS[,3],adj=5),xlim=range(SIGMA.PS[,3]),
main="",xlab="Sigma_1,3",ylab="density",lwd=2, col = 'black')
lines(density(SIGMA.prior.Sample[,3],adj=5),lwd=2, col = 'gray')
plot(density(SIGMA.PS[,5],adj=5),xlim=range(SIGMA.PS[,5]),
main="",xlab="Sigma_2,2",ylab="density",lwd=2, col = 'black')
lines(density(SIGMA.prior.Sample[,5],adj=5),lwd=2, col = 'gray')
plot(density(SIGMA.PS[,6],adj=5),xlim=range(SIGMA.PS[,6]),
main="",xlab="Sigma_2,3",ylab="density",lwd=2, col = 'black')
lines(density(SIGMA.prior.Sample[,6],adj=5),lwd=2, col = 'gray')
plot(density(SIGMA.PS[,9],adj=5),xlim=range(SIGMA.PS[,9]),
main="",xlab="Sigma_3,3",ylab="density",lwd=2, col = 'black')
```

```
lines(density(SIGMA.prior.Sample[,9],adj=5),lwd=2, col = 'gray')
```



e)

```
max.yield.density = x_grid[which.max(pred.bayesian.mean)]
max.yield.density
```

```
## [1] 5.8
```

```
x_max = c(1, max.yield.density, max.yield.density^2)
l_xmax = x_max %*% t(BETA.pp)
set.seed(1)
pred_xmax = NULL
for(i in 1:length(x_max)){
  tmp_pred_xmax = rnorm(1, l_xmax[i], sqrt(S2.b[i]))
  pred_xmax = c(pred_xmax, tmp_pred_xmax)
}
quantile(pred_xmax, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 6.729098 8.786738
```

Question 2

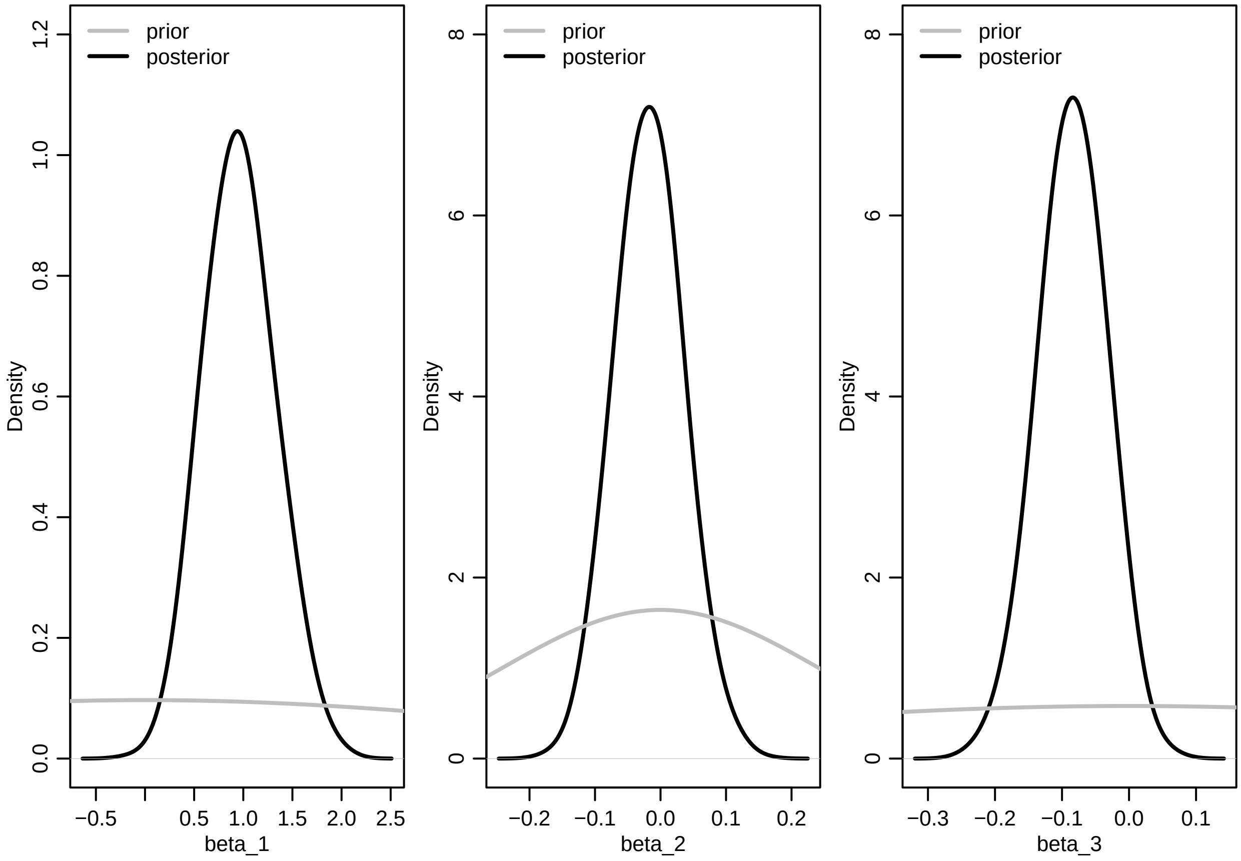
```
data_p2 = read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/prayer.dat", header = TRUE)
y = data_p2[,4]
```

```r
X = as.matrix(data_p2[,1:3])
# Rank likelihood regression
treg<-function(y,X,NSCAN=25000,seed=1) {
  set.seed(seed)
  n<-dim(X)[1] ; p<-dim(X)[2]
  iXX<-solve(t(X)%*%X) ; V<-iXX*(n/(n+1)) ; cholV<-chol(V)
  ranks<-match(y,sort(unique(y))) ; uranks<-sort(unique(ranks))
  z<-qnorm(rank(y,ties.method="random")/(n+1))
  b<-matrix(0,p,1) ; BETA<-matrix(NA,1000,p) ; Z<-matrix(NA,1000,n)
  ac<-0
  for(nscan in 1:NSCAN) {
    ###update b
    E<- V%*%( t(X)%*%z )
    b<- cholV%*%rnorm(p) + E
    ###update z
    mu<-X%*%b
    for(r in sample(uranks)) {
      ir<-(1:n)[ranks==r]
      lb<-suppressWarnings(max(z[ranks<r]))
      ub<-suppressWarnings(min(z[ranks>r]))
      z[ir]<-qnorm(
        runif( length(ir), pnorm(lb,mu[ir],1), pnorm(ub,mu[ir],1) ),
        mu[ir],1
      )
    }
    ###help mixing
    zp<-z+rnorm(1,0,n^(-1/3) )
    lhr<- sum(dnorm(zp,mu,1,log=T) - dnorm(z,mu,1,log=T) )
    if(log(runif(1))<lhr) { z<-zp ; ac<-ac+1}
    ###output
    if(nscan%%(NSCAN/1000)==0) {
      BETA[nscan/(NSCAN/1000),]<- t(b)
      Z[nscan/(NSCAN/1000),]<- z
    }
  }
  list( BETA=BETA,Z=Z )
}
treg.adhoc<-function(y,X,z=qnorm(rank(y,ties.method="random")/(length(z)+1)))
{
  ranks<-match(y,sort(unique(y))) ; uranks<-sort(unique(ranks))
  for(s in 1:10)
  {
    fit<-lm(z~-1+X) ; b<-fit$coef ; e<-fit$res
    for(r in uranks)
    {
      ir<-(1:n)[ranks==r]
      lb<-suppressWarnings(max(z[ranks<r]))
      ub<-suppressWarnings(min(z[ranks>r]))
      z[ir]<-sort(z[ir] )[rank(-e[ir]) ]
    }
  }
  z
}
```

```
rfit <- treg(y,X)
n<-dim(X)[1]
sd<-sqrt(solve(t(X)%*%X/n))
```

```
## Warning in sqrt(solve(t(X) %*% X/n)): NaNs produced
```

```
x<-seq(-5,5,length=1000)
par(mfrow=c(1,3),mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0))
plot(density(rfit$BETA[,1],adj=2),lwd=2,main="",
xlab='beta_1',col="black",ylim=c(0,1.2),ylab='Density')
lines(x, dnorm(x, 0, sd[1,1]), lwd = 2, col = 'gray')
legend('topleft',legend=c("prior","posterior"),lwd=c(2,2),col=c("gray","black"),bty="n")
plot(density(rfit$BETA[,2],adj=2),lwd=2,main="",
xlab='beta_2',col="black",ylim=c(0,8),ylab='Density')
lines(x, dnorm(x, 0, sd[2,2]), lwd = 2, col = 'gray')
legend('topleft',legend=c("prior","posterior"),lwd=c(2,2),col=c("gray","black"),bty="n")
plot(density(rfit$BETA[,3],adj=2),lwd=2,main="",
xlab='beta_3',col="black",ylim=c(0,8),ylab='Density')
lines(x, dnorm(x, 0, sd[3,3]), lwd = 2, col = 'gray')
legend('topleft',legend=c("prior","posterior"),lwd=c(2,2),col=c("gray","black"),bty="n")
```



d)

```
quantile(rfit$BETA[,1], c(0.025, 0.975))
```

```
##       2.5%     97.5%
## 0.3154128 1.6833093
```

```
quantile(rfit$BETA[,2], c(0.025, 0.975))
```

```
##        2.5%       97.5%
## -0.11285892  0.08219409
```

```
quantile(rfit$BETA[,3], c(0.025, 0.975))
```

```
##         2.5%        97.5%
## -0.184092059  0.008249275
```

The coefficient

$$\beta_1$$

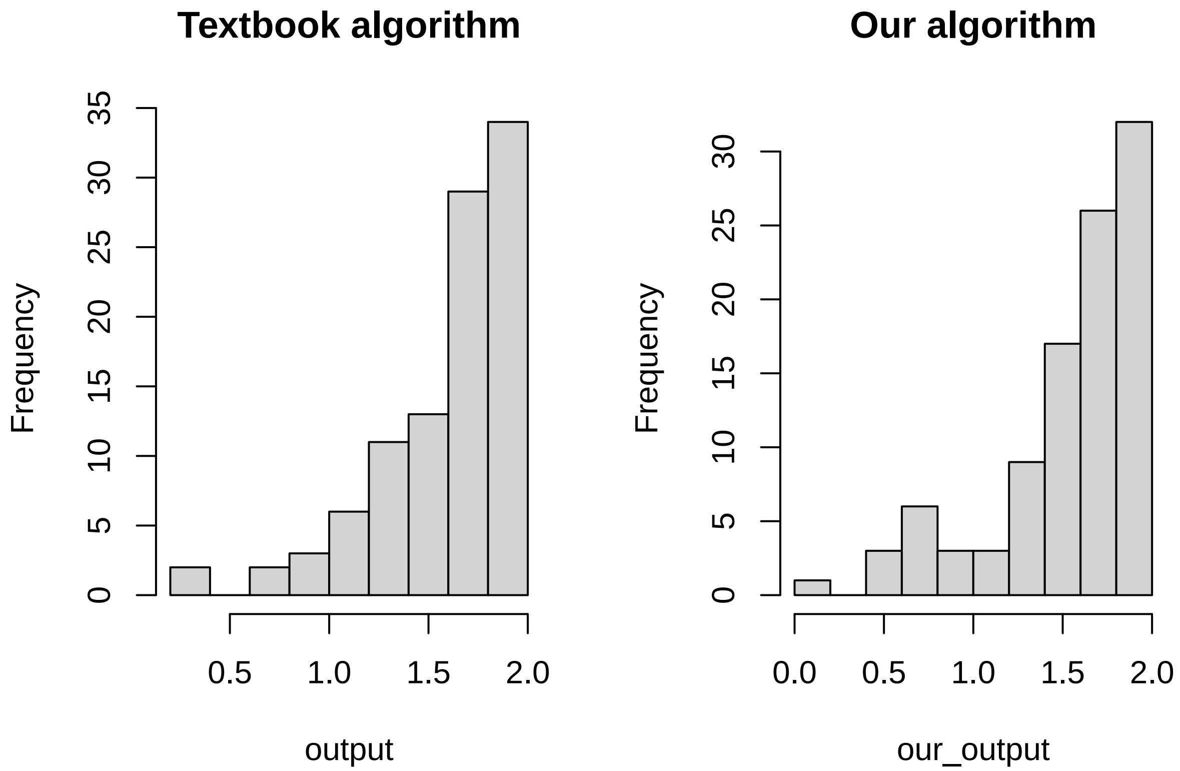
is significant, which means female pray more than male.

Question 3

```
n= 100
mu = 8
sigma2 = 3
a = 0
b = 2
# Textbook algorithm
u = runif(100, pnorm((a-mu)/sqrt(sigma2)), pnorm((b-mu)/sqrt(sigma2)))
output = mu + sqrt(sigma2)*qnorm(u)
set.seed(1)
# Our naive algorithm
i = 0
iter = 0
our_output = NULL
while(i < n){
  iter = iter + 1
  x = rnorm(1, mu, sqrt(sigma2))
  if(x > a && x < b){
    our_output = c(our_output, x)
    i = i+1
  }
}
print(paste('Our algorithm iterates', iter, 'times.'))
```

```
## [1] "Our algorithm iterates 343578 times."
```

```
par(mfrow = c(1, 2))
hist(output, main = 'Textbook algorithm')
hist(our_output, main = 'Our algorithm')
```



The textbook algorithm significantly outperform the algorithm implemented here.

Question 4

```
library(sbgcop)
```

```
##
## Attaching package: 'sbgcop'

## The following object is masked _by_ '.GlobalEnv':
##
##     rwish
```

```
load("socmob.RData")
data_p4 = socmob
n = dim(data_p4)[1]
p = dim(data_p4)[2]
nu0 = p + 2
S0 =(p+2)*diag(p)
n_iter = 1000

res_copula <- sbgcop.mcmc(
  Y = data_p4,
  S0 = S0,
  n0 = nu0,
  nsamp = 10000,
  odens = 10,
  seed = 1,
  verb = FALSE
)
cov_all= res_copula$C.psamp
apply(cov_all, c(1,2), mean)
```

```
##                INCOME      DEGREE    CHILDREN    PINCOME    PDEGREE   PCHILDREN
## INCOME     1.00000000  0.47070246  0.28694099  0.1260760  0.1697058 -0.04924544
## DEGREE     0.47070246  1.00000000 -0.04059447  0.2014234  0.4553500 -0.19899551
## CHILDREN   0.28694099 -0.04059447  1.00000000 -0.1474823 -0.2470654  0.21871297
## PINCOME    0.12607599  0.20142336 -0.14748227  1.0000000  0.4255160 -0.20717398
## PDEGREE    0.16970583  0.45534996 -0.24706536  0.4255160  1.0000000 -0.27950214
## PCHILDREN -0.04924544 -0.19899551  0.21871297 -0.2071740 -0.2795021  1.00000000
## AGE        0.32376685  0.04794096  0.57404790 -0.1240546 -0.2194395  0.11530131
##                   AGE
## INCOME     0.32376685
## DEGREE     0.04794096
## CHILDREN   0.57404790
## PINCOME   -0.12405462
## PDEGREE   -0.21943953
## PCHILDREN  0.11530131
## AGE        1.00000000
```