Homework8_STA602

ElenaW.

10/26/2021

```
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)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

7.3

7.3.a)

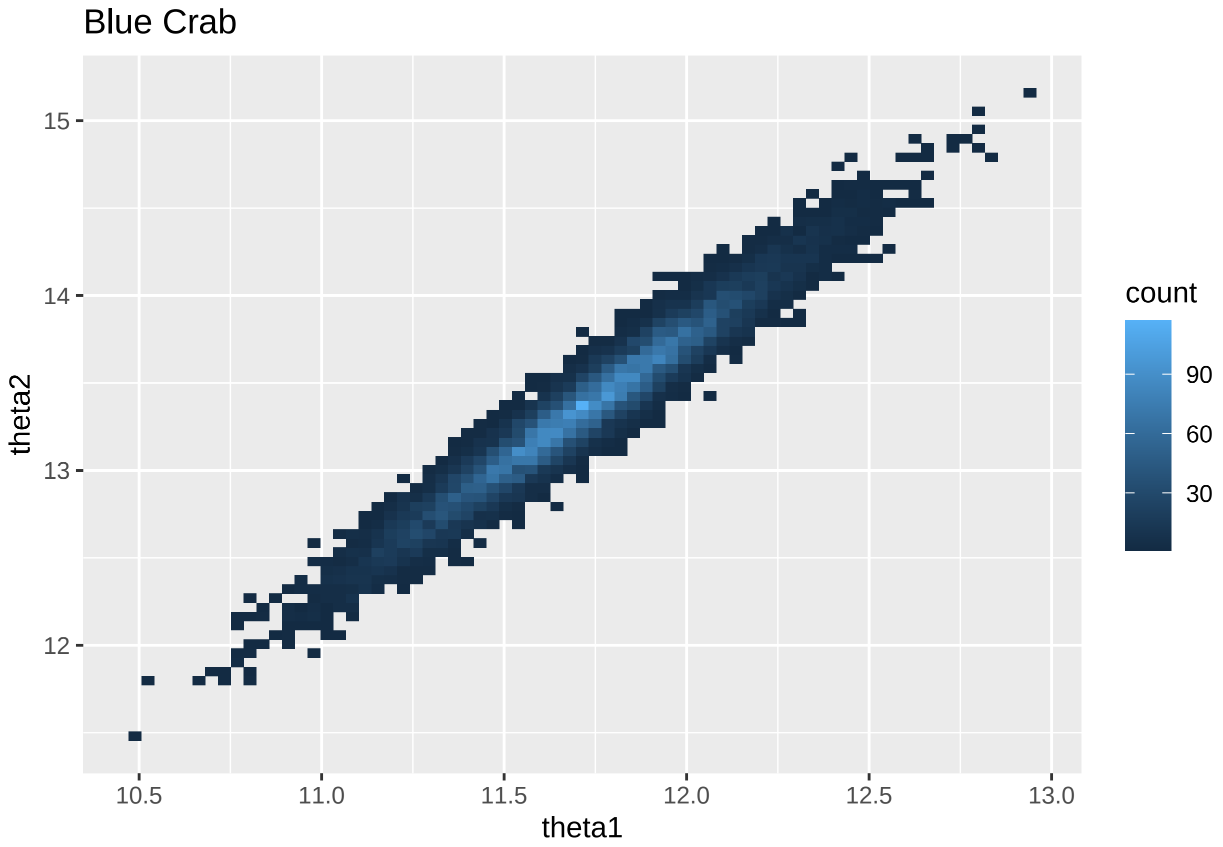
```
blue = read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/bluecrab.dat")
orange = read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/orangecrab.dat")
```

```
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 = 4
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")

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))
}
```

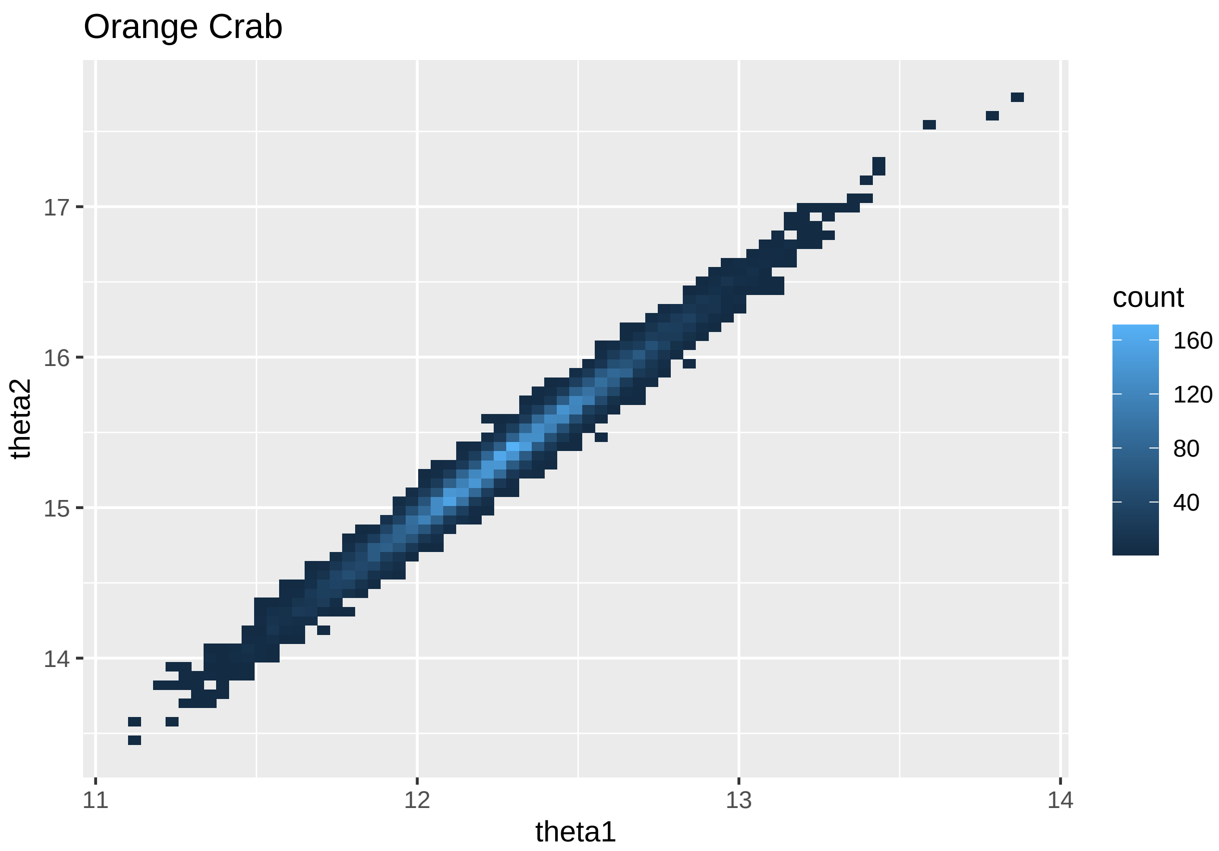
```
sim_blue = Gibbs(blue) # blue crab simulation

sim_orange = Gibbs(orange) # orange crab simulation
```

b)

```
par(mfrow = c(1,2))
theta_blue = data.frame(sim_blue$theta)
theta_orange = data.frame(sim_orange$theta)
ggplot(theta_blue, aes(x = theta1, y = theta2))+
  geom_bin2d(bins = 70)+
  labs(x = "theta1",
       y = "theta2",
       title = "Blue Crab")
```



```
ggplot(theta_orange, aes(x = theta1, y = theta2))+
  geom_bin2d(bins = 70)+
  labs(x = "theta1",
       y = "theta2",
       title = "Orange Crab")
```



```
mean(theta_blue$theta1 > theta_orange$theta1)
```

```
## [1] 0.0998
```

```
mean(theta_blue$theta2 > theta_orange$theta2)
```

```
## [1] 0.0014
```

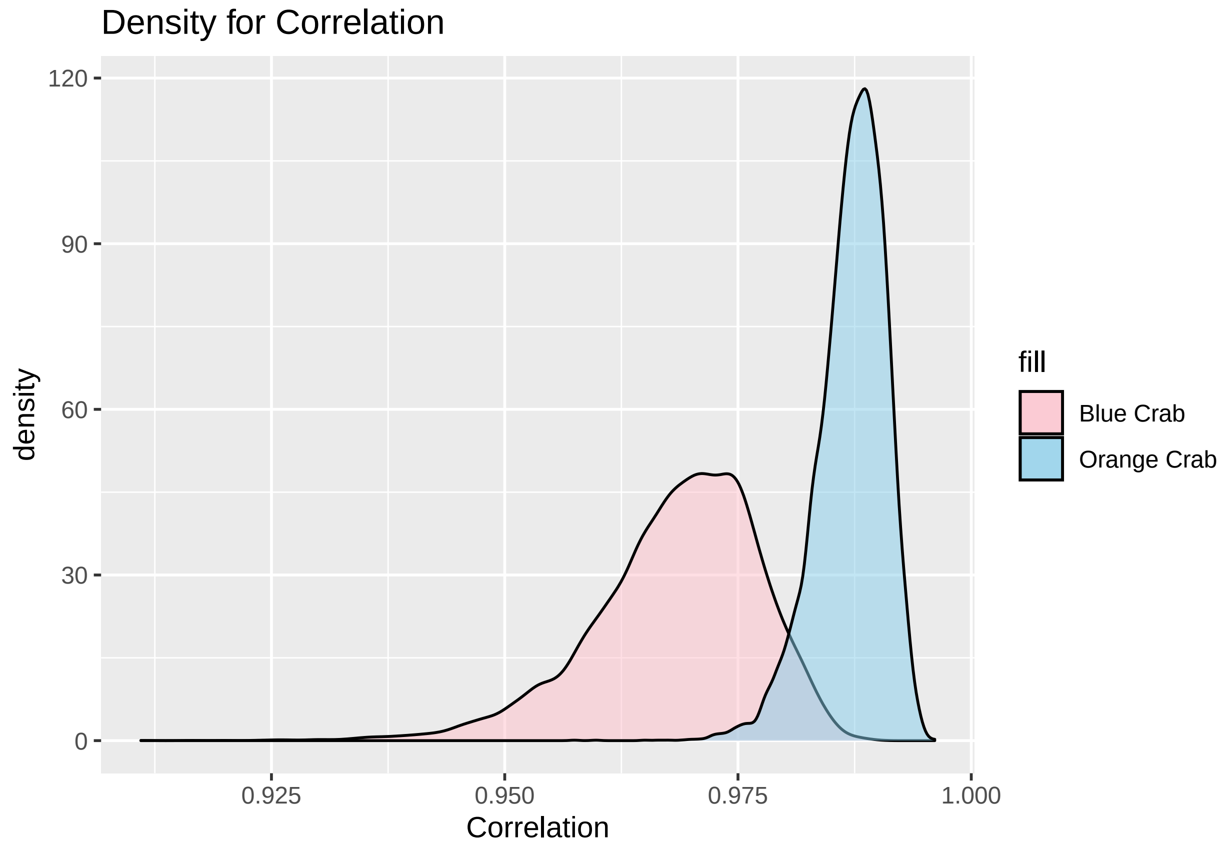
From the plot and the calculation above, we could see that theta1 and theta2 for orange crab tends to be bigger than for the blue crab.

c)

```
correlation = function(matrix){
  correlation = matrix[2]/sqrt(matrix[1]*matrix[4])
  return(correlation)
}
```

```
sigma_blue = sim_blue$sigma
cor_blue = apply(sigma_blue, 1, correlation)

sigma_orange = sim_orange$sigma
```

```
cor_orange = apply(sigma_orange,1,correlation)

ggplot()+
  geom_density(aes(x = cor_blue, fill = "pink"), alpha = 0.5)+
  geom_density(aes(x = cor_orange, fill = "skyblue"), alpha = 0.5)+
  scale_fill_manual(values = c("pink", "skyblue"),
                    labels = c("Blue Crab",
                               "Orange Crab"))+
  labs(x = "Correlation",
       title = "Density for Correlation")
```



```
mean(cor_blue < cor_orange)
```

```
## [1] 0.9897
```

The orange crab species has much higher correlation between its two measurements than the blue one.

7.4

a)

```
age = read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/agehw.dat")
age = data.frame(age)
age = age %>%
  row_to_names(row_number = 1)
age = as.data.frame(lapply(age, as.numeric))
```

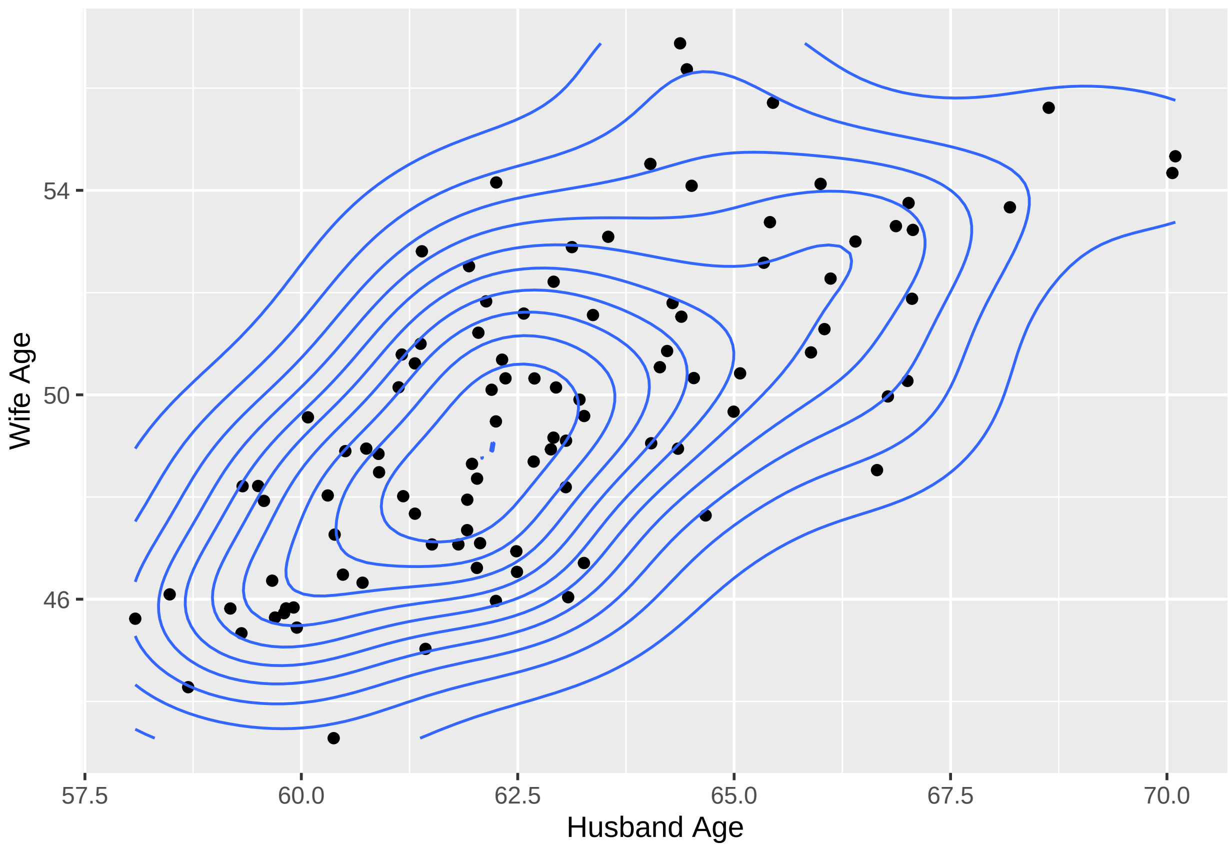
Since the range of men age falls between 22 and 79 and the range of women women range falls between 17 and 72, which means that they both fall from 17 - 79. Thus I would take the average age from 17 - 79, which is 48 and I'm confident that their average age would be really close but husband would be higher than wife. Hence, I would take mu_0 = (50,48)^t. From my prior knowledge, I guess the variance of both age is 150, the sigma (covariance matrix) chosen is (150,100,100,150). I would also use the same covariance matrix for S0. nu_0 I set is p + 10, since we have some idea about the value of sigma but not a lot, so we want to choose a not really weak prior.

```
mu_0 = c(50,48)
lambda_0 = rbind(c(150,100),c(100,150))
s_0 = rbind(c(150,100),c(100,150))
p = 2
nu_0 = p + 10
```

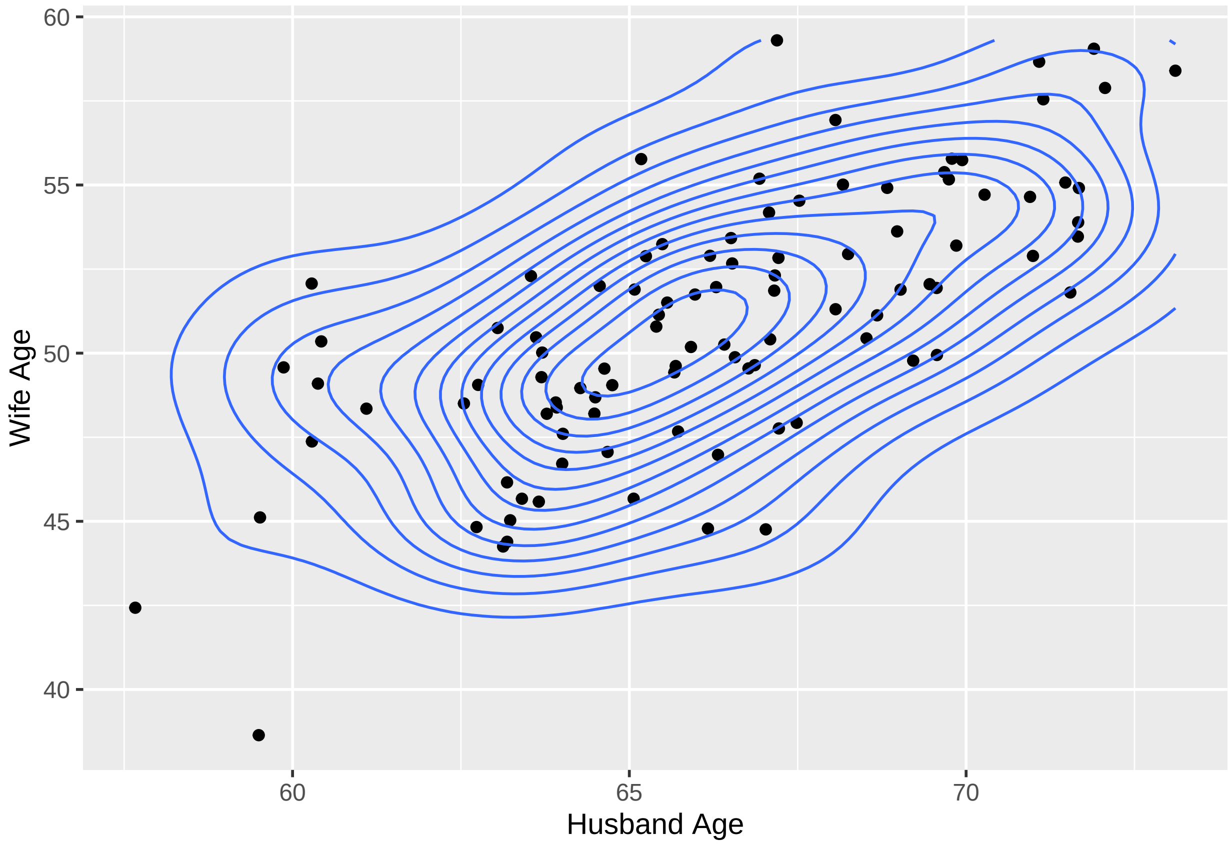
b)

```
set.seed(0)
pred = c()
for (i in 1:3){
  theta = rmvnorm(1, mu_0, lambda_0)
  sigma = riwish(nu_0, s_0)
  y_pred = rmvnorm(100,theta, sigma)
  pred = cbind(pred, y_pred)
}
```

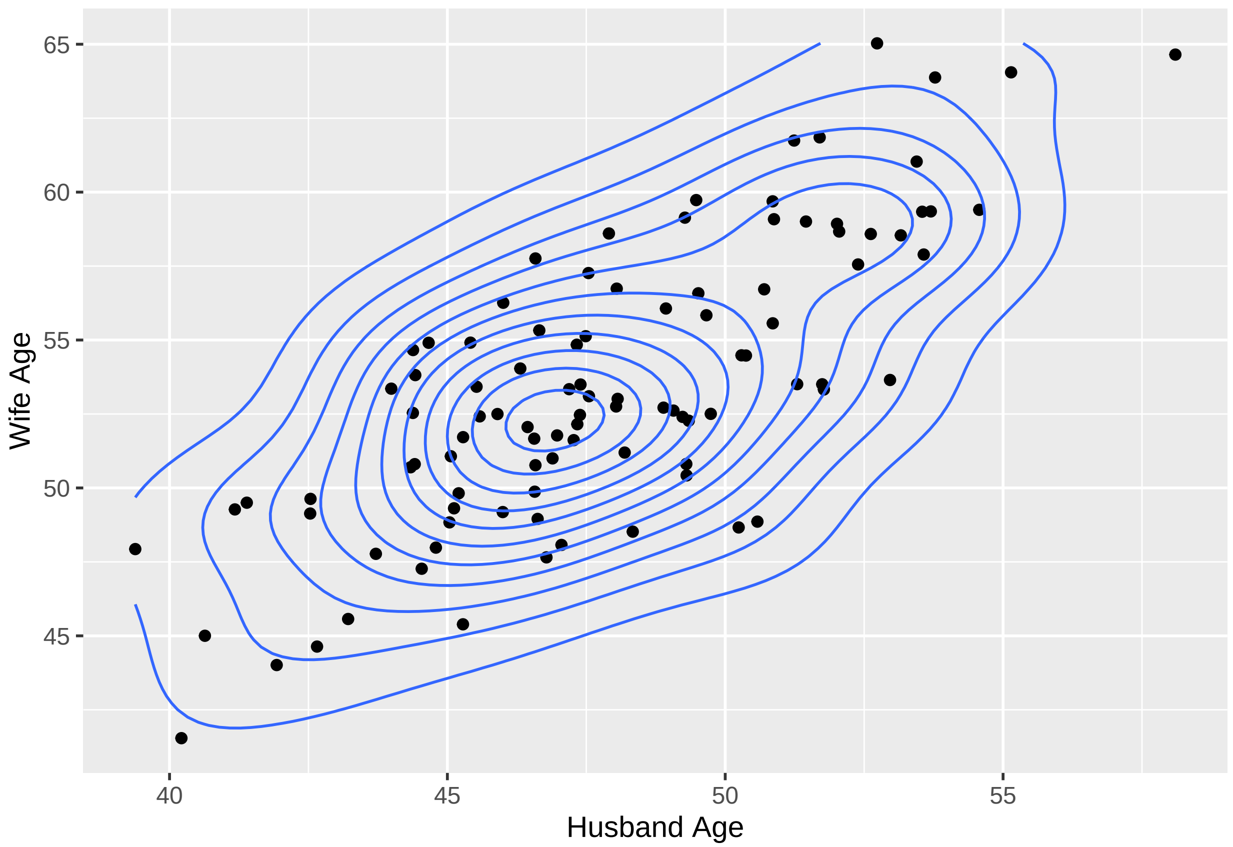
```
par(mfrow = c(1,3))
ggplot()+
 geom_point(aes(x = pred[,1], y = pred[,2]))+
 geom_density_2d(aes(x = pred[,1], y = pred[,2]))+
 labs(x = "Husband Age",
 y = "Wife Age")
```



```
ggplot()+
 geom_point(aes(x = pred[,3], y = pred[,4]))+
 geom_density_2d(aes(x = pred[,3], y = pred[,4]))+
 labs(x = "Husband Age",
 y = "Wife Age")
```



```
ggplot()+
 geom_point(aes(x = pred[,5], y = pred[,6]))+
 geom_density_2d(aes(x = pred[,5], y = pred[,6]))+
 labs(x = "Husband Age",
 y = "Wife Age")
```



Since from the scatterplots above, although the first two plots are kind of away from the prior but we could see that the center of husband age and wife age in the third one is really close.

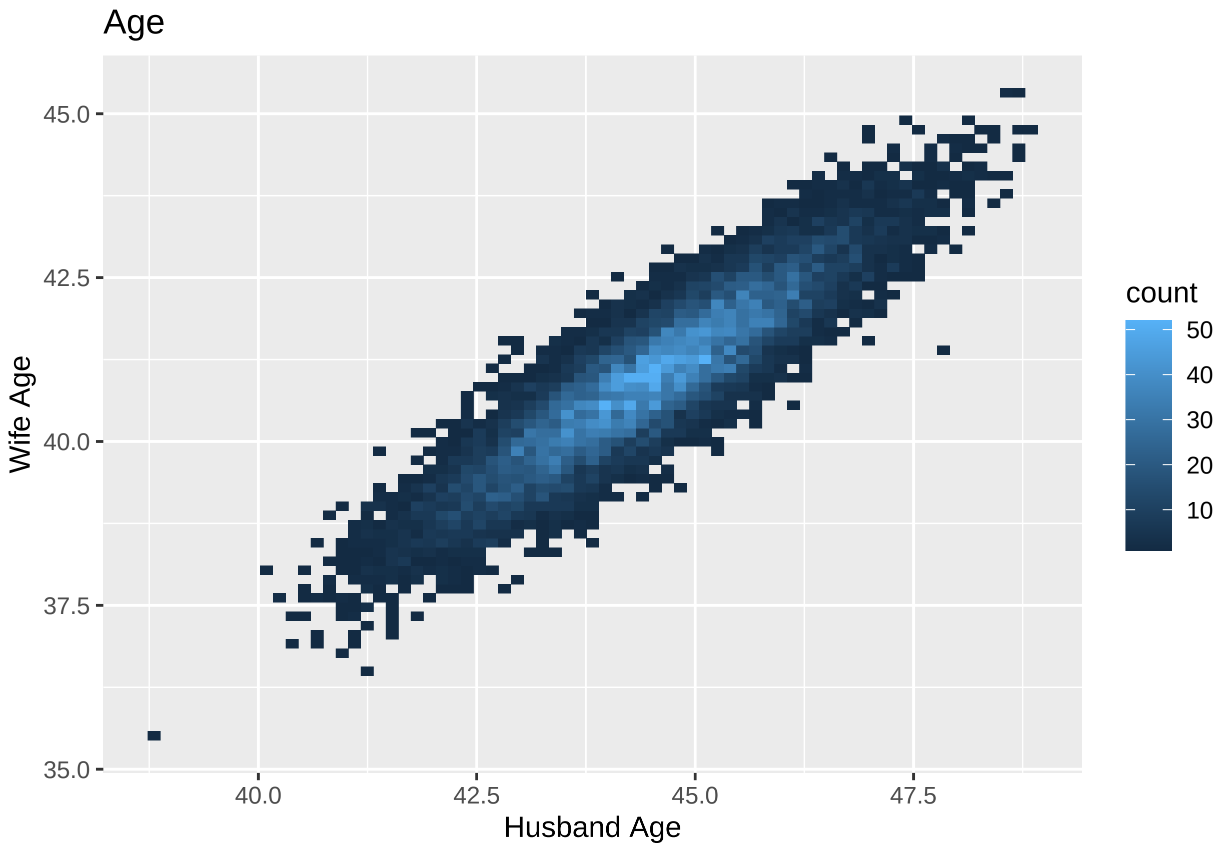
c)

```
set.seed(0)
Gibbs_age = function(y){
n = nrow(y) # sample size
p = ncol(y) # dimensionality

# prior for theta
mu.0 = c(50,48)
lambda.0 = rbind(c(150,100),c(100,150))

# prior for sigma
nu.0 = p + 10
S0 = rbind(c(150,100),c(100,150))

ybar = colMeans(y)
nu.n = nu.0 + n

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

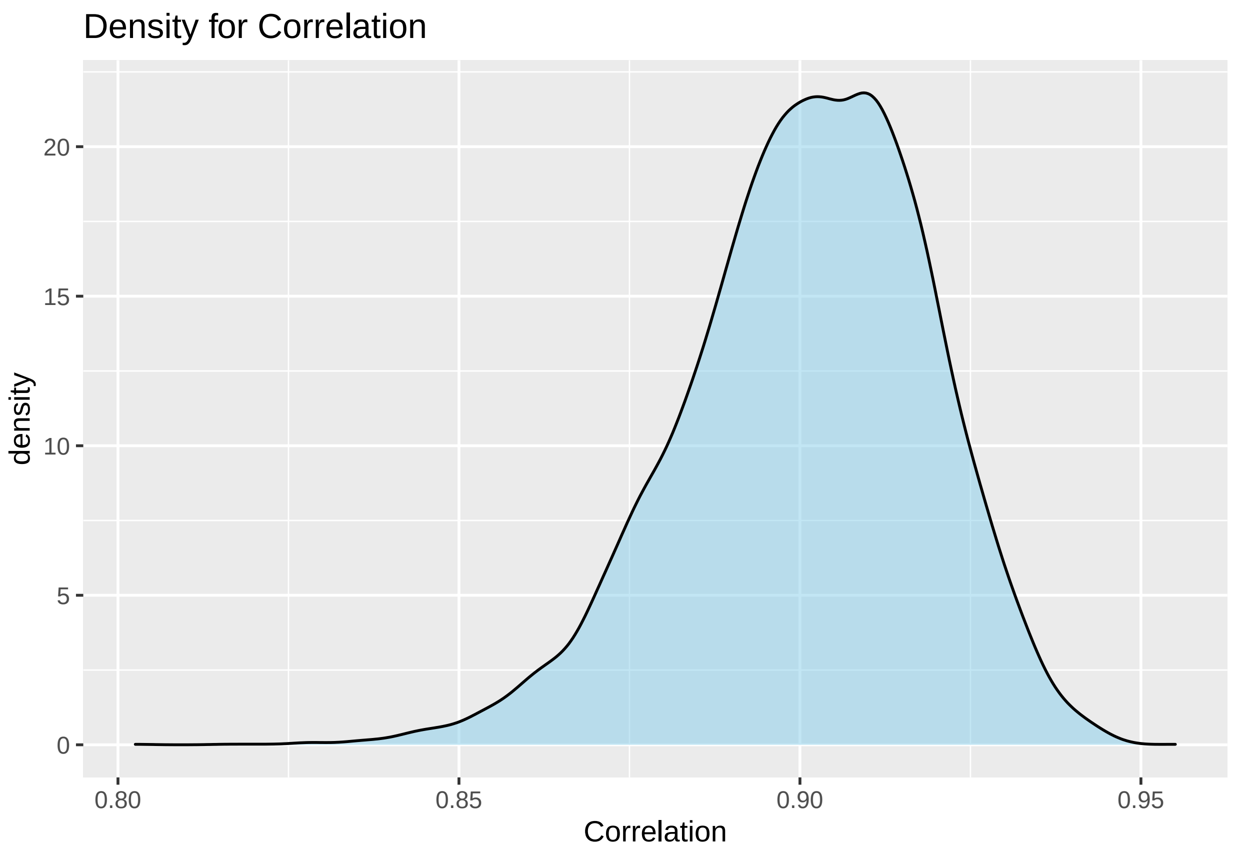
```r
THETA = matrix(NA, nrow = niter, ncol = p) # matrix for storing the draws for theta
colnames(THETA) = c("theta1", "theta2")

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 = 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))
}
```

```r
mcmc_age = Gibbs_age(age)
```

```r
theta_age = data.frame(mcmc_age$theta)
ggplot(theta_age, aes(x = theta1, y = theta2))+
 geom_bin2d(bins = 70)+
 labs(x = "Husband Age",
 y = "Wife Age",
 title = "Age")
```



```r
cor_y = mcmc_age$sigma
correlation_y = apply(cor_y,1, correlation)

ggplot()+
 geom_density(aes(x=correlation_y), fill = "skyblue", alpha = 0.5)+
 labs(x = "Correlation",
 title = "Density for Correlation")
```



```
quantile(theta_age$theta1, probs = c(0.025,0.975))
```

```
##     2.5%    97.5%
## 41.93330 47.02424
```

```
quantile(theta_age$theta2, probs = c(0.025,0.975))
```

```
##     2.5%    97.5%
## 38.55864 43.37389
```

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

```
##      2.5%     97.5%
## 0.8619482 0.9319960
```

d)

i) the full condition distribution of sigma is in paper

```
set.seed(0)
Gibbs_jeffrey = function(y){
n = nrow(y) # sample size
p = ncol(y) # dimensionality
# prior for theta
mu.0 = c(50,48)
lambda.0 = rbind(c(150,100),c(100,150))

# prior for sigma
nu.0 = p + 10
S0 = cov(y)

ybar = colMeans(y)
nu.n = 1 + 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")

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)

  ## Save the current iteration
  THETA[t,] <- THETA.curr
  SIGMA[t,] <- SIGMA.curr
}
return(list(theta = THETA,sigma = SIGMA))
}
```

```
mcmc_jeffrey = Gibbs_jeffrey(age)
theta_jeffrey = data.frame(mcmc_jeffrey$theta)
sigma_jeffrey = mcmc_jeffrey$sigma
cor_jeffrey = apply(sigma_jeffrey, 1, correlation)
quantile(theta_jeffrey$theta1,probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 41.71992 47.08826
```

```
quantile(theta_jeffrey$theta2,probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 38.35226 43.44030
```

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

```
##      2.5%     97.5%
## 0.8622640 0.9348092
```

ii) Deviation is on the paper

```
set.seed(0)
Gibbs_unit = function(y){
n = nrow(y) # sample size
p = ncol(y) # dimensionality

# prior for theta
mu.0 = c(50,48)
lambda.0 = rbind(c(150,100),c(100,150))

# prior for sigma
nu.0 = p + 10
S0 = cov(y)

ybar = colMeans(y)
nu.n = 1 + n + p

# 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")

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+1)
  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*(n+1)/n)

  ## Save the current iteration
  THETA[t,] <- THETA.curr
  SIGMA[t,] <- SIGMA.curr
}
return(list(theta = THETA,sigma = SIGMA))
}
```

```
mcmc_unit = Gibbs_unit(age)
theta_unit = data.frame(mcmc_unit$theta)
sigma_unit = mcmc_unit$sigma
cor_unit = apply(sigma_unit, 1, correlation)
quantile(theta_unit$theta1,probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 41.74741 47.06081
```

```
quantile(theta_unit$theta2,probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 38.37553 43.40892
```

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

```
##      2.5%     97.5%
## 0.8627951 0.9345216
```

```
set.seed(0)
Gibbs_diffuse = function(y){
n = nrow(y) # sample size
p = ncol(y) # dimensionality

# prior for theta
mu.0 = c(0,0)
lambda.0 = rbind(c(10^5, 0), c(0,10^5))

# prior for sigma
nu.0 = 3
S0 = rbind(c(1000,0), c(0, 1000))

ybar = colMeans(y)
nu.n = 3 + 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")

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))
}
```

```
mcmc_diffuse = Gibbs_diffuse(age)
theta_diffuse = data.frame(mcmc_diffuse$theta)
sigma_diffuse = mcmc_diffuse$sigma
cor_diffuse = apply(sigma_diffuse, 1, correlation)
quantile(theta_diffuse$theta1,probs = c(0.025, 0.975))
```

iii)

```
##     2.5%    97.5%
## 41.66499 47.13190
```

```
quantile(theta_diffuse$theta2,probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 38.28851 43.48149
```

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

```
##      2.5%     97.5%
## 0.7939957 0.8999906
```

e)

The confidence intervals from d) and c) are really close whatever is for mean or correlation, which means that my prior information is helpful in estimating theta and sigma. If the sample size is just 25, the confidence interval in part c) would get wider than in part d), since my prior degree of freedom is 10 + p + n, which is higher than other prior in part d) and my prior would have more influence on the posterior distribution than in larger sample size, and we would have larger variance.