

STA 602. HW10

Yicheng Shen

08 November, 2022

[1] 7.4

(a) We could formulate a combination of Normal-Inverse-Wishart prior in this case.

Specifically, for average husband and wife ages, I think they should be around 50 and 45, with big uncertainty in my belief since I am not sure. I also think there should be strong correlation between the ages of husband and wife. Therefore:

$$\begin{aligned}\boldsymbol{\theta} = (\theta_h, \theta_w)^\top &\sim N_2(\boldsymbol{\mu}_0, \Lambda_0) \\ &= N\left(\begin{bmatrix} 50 \\ 45 \end{bmatrix}, \begin{bmatrix} 100 & 70 \\ 70 & 100 \end{bmatrix}\right)\end{aligned}$$

For the covariance matrix, an Inverse-Wishart prior can induce semi-conjugacy. Thus, I put prior degree of freedom at 5, and use prior covariance matrix similar to Λ_0 :

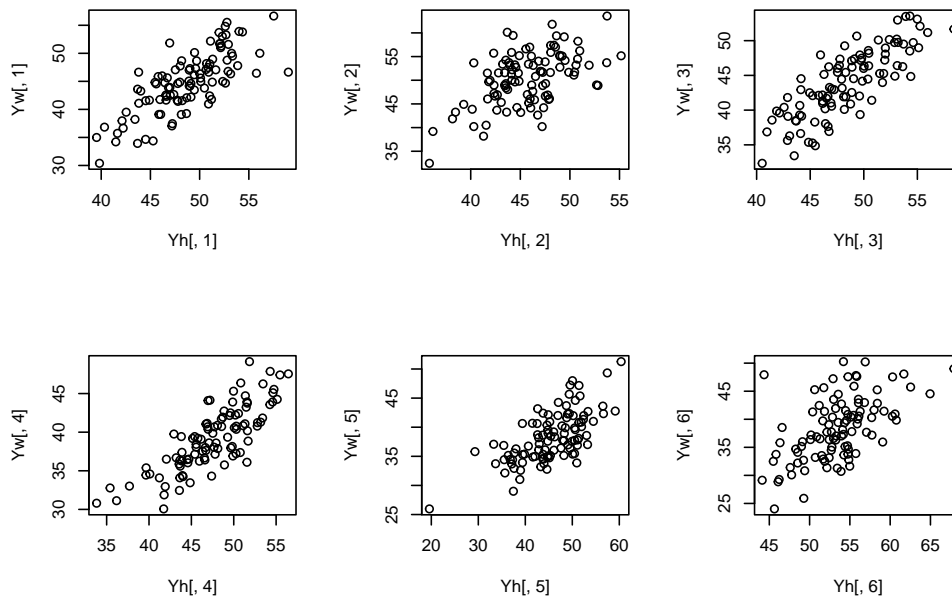
$$\begin{aligned}\Sigma &\sim IW(\nu_0, S_0) \\ &= IW(5, \begin{bmatrix} 100 & 70 \\ 70 & 100 \end{bmatrix})\end{aligned}$$

(b) My simulated data is drawn below, and they look reasonable to me.

```
set.seed(2546)

mu.0 = c(50,45)
Lambda.0 = S0 = matrix(c(100, 70, 70, 100), nrow = 2)
nu.0 = 5

n = 100
Yh = Yw = matrix(NA, nrow = n, ncol = 6)
for(i in 1:6) {
  theta = mvtnorm::rmvnorm(1, mu.0, Lambda.0)
  Sigma = MCMCpack::riwish(v = nu.0, S = S0)
  Y.sim = mvtnorm::rmvnorm(n, theta, Sigma)
  Yh[, i] = Y.sim[, 1]
  Yw[, i] = Y.sim[, 2]
}
par(mfrow = c(2, 3))
plot(Yh[,1], Yw[,1])
plot(Yh[,2], Yw[,2])
plot(Yh[,3], Yw[,3])
plot(Yh[,4], Yw[,4])
plot(Yh[,5], Yw[,5])
plot(Yh[,6], Yw[,6])
```



(c)

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

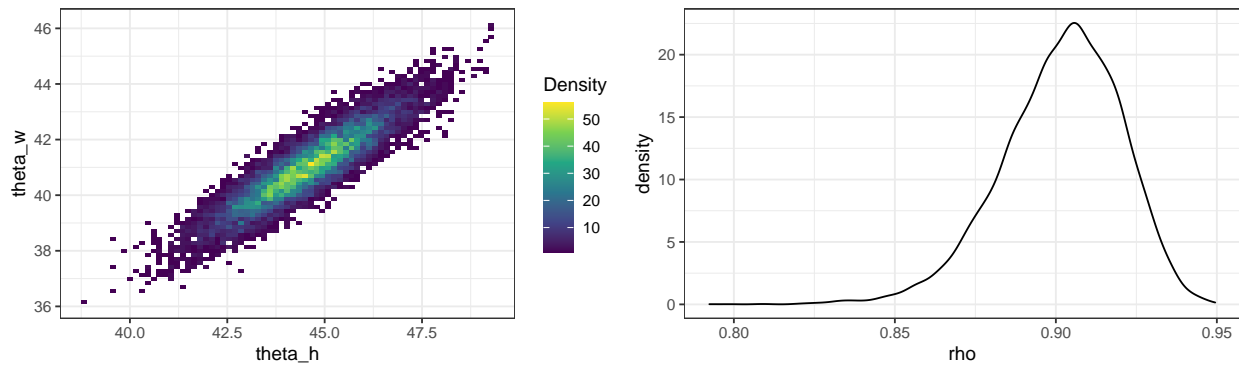
```
y <- as.matrix(agehw)
ybar <- apply(y, 2, mean)
nu.n <- nu.0 + n
niter <- 10000 # total number of iterations
THETA <- matrix(NA, nrow = niter, ncol = 2) # matrix for storing the draws for theta
colnames(THETA) <- c("theta1", "theta2")
THETA.init <- ybar # Initial values set to sample mean
THETA.curr <- THETA.init # the theta value at current iteration
SIGMA <- matrix(NA, nrow = niter, ncol = 2 * 2) # matrix for storing the draws for Sigma
colnames(SIGMA) <- c("sigma11", "sigma12", "sigma21", "sigma22")
SIGMA.init <- cov(y) # initial value set to sample covariance
SIGMA.curr <- SIGMA.init # the Sigma value at current iteration
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)
  ## 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
}
```

The joint density plot and marginal density of correlation plot are plotted below.

```

a <- ggplot(data.frame(THETA), aes(x = theta1, y = theta2)) +
  labs(x = "theta_h", y = "theta_w", fill = "Density") +
  geom_bin2d(bins = 70) + scale_fill_continuous(type = "viridis")
b <- ggplot(as.data.frame(SIGMA)) +
  geom_density(aes(x = sigma12 / sqrt(sigma11) / sqrt(sigma22))) + labs(x = "rho")
grid.arrange(a, b, nrow = 1)

```



The 95% CI for θ_h , θ_w and correlation coefficient are shown below.

```
quantile(THETA[,1], c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 41.89429 47.17474
```

```
quantile(THETA[,2], c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 38.55648 43.44969
```

```
quantile(SIGMA[,2] / sqrt(SIGMA[,1] * SIGMA[,4]), c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 0.8609600 0.9330098
```

(d) This problem is divided into three parts:

i. For Jeffery's prior, which is

$$P(\theta, \Sigma)_J \propto \Sigma^{-(p+2)/2}$$

$$P(\theta)_J \propto 1; P(\Sigma)_J \propto \Sigma^{-(p+2)/2}$$

From Ex 7.1

$$\theta | \Sigma, \mathbf{y} \sim MVN(\bar{y}, \Sigma/n)$$

$$\Sigma | \theta, \mathbf{y} \sim IW(n+1, S_\theta)$$

```

THETA <- matrix(NA, nrow = niter, ncol = 2)
colnames(THETA) <- c("theta1", "theta2")
THETA.init <- ybar
THETA.curr <- THETA.init
SIGMA <- matrix(NA, nrow = niter, ncol = 2 * 2)
colnames(SIGMA) <- c("sigma11", "sigma12", "sigma21", "sigma22")
SIGMA.init <- cov(y)
SIGMA.curr <- SIGMA.init

for (t in 1:niter) {
  ## Update theta
  THETA.curr <- rmvnorm(1, mean = ybar, sigma = SIGMA.curr/n)
  ## Update Sigma
  S.theta <- (t(y) - c(THETA.curr)) %*% t(t(y) - c(THETA.curr))
  SIGMA.curr <- riwish(v = n+1, S = S.theta)
  ## Save the current iteration
  THETA[t, ] <- THETA.curr
  SIGMA[t, ] <- SIGMA.curr
}

```

	2.5%	97.5%
θ_h	41.714	47.064
θ_w	38.357	43.354
correlation	0.860	0.935

- ii. For the unit information prior, also borrow from my last homework

$$\begin{aligned}
p(\theta, \Sigma | Y) &\propto \text{Inverse-Wishart}(p+1+n, (1+n)S) \times \text{MVN}(\bar{\mathbf{y}}, \Sigma/(n+1)) \\
\theta | \Sigma, \mathbf{y} &\sim \text{MVN}(\bar{\mathbf{y}}, \Sigma/(n+1)) \\
\Sigma | \theta, \mathbf{y} &\sim \text{IW}(p+1+n, (1+n)S) \\
\text{where } S &= \sum_i (y_i - \bar{y})^\top (y_i - \bar{y}) / n
\end{aligned}$$

```

THETA <- matrix(NA, nrow = niter, ncol = 2)
colnames(THETA) <- c("theta1", "theta2")
THETA.init <- ybar
THETA.curr <- THETA.init
SIGMA <- matrix(NA, nrow = niter, ncol = 2 * 2)
colnames(SIGMA) <- c("sigma11", "sigma12", "sigma21", "sigma22")
SIGMA.init <- cov(y)
SIGMA.curr <- SIGMA.init

for (t in 1:niter) {
  ## Update theta
  THETA.curr <- rmvnorm(1, mean = ybar, sigma = (SIGMA.curr/(n+1)) )
  ## Update Sigma
  S <- (t(y) - ybar)%*%t(t(y) - ybar)
  SIGMA.curr <- riwish(v = n+2+1, S = S / n * (n+1) )
  ## Save the current iteration
  THETA[t, ] <- THETA.curr
  SIGMA[t, ] <- SIGMA.curr
}

```

	2.5%	97.5%
θ_h	41.777	47.102
θ_w	38.420	43.415
correlation	0.862	0.934

i. For diffuse prior

```
mu.0 = c(0,0)
Lambda.0 = matrix(c(100000, 0, 0, 100000), nrow = 2)
S0 = matrix(c(1000, 0, 0, 1000), nrow = 2)
nu.0 = 3
nu.n <- nu.0 + n
niter <- 10000 # total number of iterations
THETA <- matrix(NA, nrow = niter, ncol = 2) # 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 = 2 * 2) # matrix for storing the draws for Sigma
colnames(SIGMA) <- c("sigma11", "sigma12", "sigma21", "sigma22")
SIGMA.init <- cov(y) # intial value set to sample covariance
SIGMA.curr <- SIGMA.init # the Sigma value at current iteration
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)
  ## 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
}
```

	2.5%	97.5%
θ_h	41.681	47.121
θ_w	38.358	43.449
correlation	0.793	0.900

(e) In comparison, my proposed priors are very similar to Jeffery's prior and unit information priors. Only diffuse prior seems to be a bit different when estimating the correlation.

I think this is mainly due to the fact we have a big sampled data of $n = 100$. If we have a much smaller n , like 25, the 95% CI of different priors will be more different, since priors can affect the posterior much more now.

[2] 7.5

(a) $\hat{\theta}_A = 24.20049$, $\hat{\sigma}_A^2 = 4.09280$, $\hat{\theta}_A = 24.805349$, $\hat{\sigma}_A^2 = 4.691578$, and $\hat{\rho} = 0.6164509$

```
interexp <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/interexp.dat", header = T)
c(mean(interexp$yA, na.rm = T), var(interexp$yA, na.rm = T))
```

```
## [1] 24.20049 4.09280
```

```
c(mean(interexp$yB, na.rm = T), var(interexp$yB, na.rm = T))
```

```
## [1] 24.805349 4.691578
```

```
interexp %>% na.omit() %>% cor()
```

```
##           yA           yB
## yA 1.0000000 0.6164509
## yB 0.6164509 1.0000000
```

(b) The paired sample t-test suggests there are statistically significant difference and the 95% confidence interval for $\theta_A - \theta_B$ is $(-0.9850730, -0.2383347)$. So it suggests that θ_B is probably greater than θ_A

```
hat_theta_A <- mean(interexp$yA, na.rm = T)
hat_theta_B <- mean(interexp$yB, na.rm = T)
hat_s2_A <- var(interexp$yA, na.rm = T)
hat_s2_B <- var(interexp$yB, na.rm = T)
hat_rho <- (interexp %>% na.omit() %>% cor())[1,2]

interexp_imputed <- interexp %>%
  mutate(yB = ifelse(is.na(yB), hat_theta_B + (yA - hat_theta_A) * hat_rho * sqrt(hat_s2_B / hat_s2_A), yB)) %>%
  mutate(yA = ifelse(is.na(yA), hat_theta_A + (yB - hat_theta_B) * hat_rho * sqrt(hat_s2_A / hat_s2_B), yA))

t.test(interexp_imputed$yA, interexp_imputed$yB, paired = TRUE)
```

```
##
## Paired t-test
##
## data: interexp_imputed$yA and interexp_imputed$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
```

(c) Here I choose to use Jeffery's prior for easier forms of prior and posterior.

```
y.original <- interexp

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

```

niter <- 8000 # total number of iterations
nburnin <- 1000 # 1000 burn-in steps
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)
# initialize y by filling in the NAs with the corresponding column means
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 <- nu.0 + n
THETA <- matrix(NA,nrow=niter,ncol=p) # matrix for storing the draws for theta
colnames(THETA) <- c("thetaA","thetaB")
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("sigma11","sigma12","sigma21","sigma22")
SIGMA.init <- cov(y) # initial value set to sample covariance
SIGMA.curr <- SIGMA.init # the Sigma value at current iteration

set.seed(99119)
for (t in 1:niter) {
  ## Update theta
  THETA.curr <- rmvnorm(1, mean = ybar, sigma = SIGMA.curr/n)
  ## Update Sigma
  S.theta <- (t(y) - c(THETA.curr)) %*% t(t(y) - c(THETA.curr))
  SIGMA.curr <- riwish(v = n + 1, 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.
        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
}

```

The posterior mean of $\theta_A - \theta_B$ is very similar to the above. The 95% CI which seems to be a wider interval compared with paired t-test and also includes zero (because the Bayesian approach considers uncertainty quantification more properly). Based on this 95% CI, we are now concluding that y_A does not necessarily smaller than y_B .

```
mean(THETA[,1] - THETA[,2])
```

```
## [1] -0.6233591
```

```
quantile(THETA[,1] - THETA[,2], c(0.025, 0.975))
```

```
##          2.5%          97.5%  
## -1.28877512  0.06153657
```

[3] 7.6

(a)

```
azdiabetes <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/azdiabetes.dat", header = T)  
azdiabetes_Y <- azdiabetes %>% filter(diabetes == "Yes") %>% dplyr::select(-diabetes)  
azdiabetes_N <- azdiabetes %>% filter(diabetes == "No") %>% dplyr::select(-diabetes)
```

First we draw 10000 samples for diabetics group

```
n <- nrow(azdiabetes_Y)  
ybar <- apply(azdiabetes_Y, 2, mean)  
mu.0 <- ybar  
Sigma <- Lambda.0 <- S0 <- cov(azdiabetes_Y)  
p <- 7  
nu0 <- 9  
  
THETA_d = SIGMA_d = NULL  
  
for(t in 1:10000){  
  
  Lambda.n <- solve(solve(Lambda.0) + n * solve(Sigma))  
  mu.n <- Lambda.n %*% (solve(Lambda.0) %*% mu.0 + n * solve(Sigma) %*% ybar)  
  theta <- mvtnorm::rmvnorm(1, mu.n, Lambda.n)  
  
  Sn <- S0 + (t(azdiabetes_Y) - c(theta)) %*% t( t(azdiabetes_Y) - c(theta))  
  Sigma <- solve( rWishart(1, nu0 + n, solve(Sn))[, , 1])  
  
  THETA_d <- rbind(THETA_d, theta)  
  SIGMA_d <- rbind(SIGMA_d, c(Sigma))  
}
```

Then we draw 10000 samples for non-diabetics group

```
n <- nrow(azdiabetes_N)  
ybar <- apply(azdiabetes_N, 2, mean)  
mu.0 <- ybar  
Sigma <- Lambda.0 <- S0 <- cov(azdiabetes_N)  
p <- 7
```



```
nu0 <- 9
```

```
THETA_n = SIGMA_n = NULL
```

```
for(t in 1:10000){
```

```
  Lambda.n <- solve(solve(Lambda.0) + n * solve(Sigma))
```

```
  mu.n <- Lambda.n %*% (solve(Lambda.0) %*% mu.0 + n * solve(Sigma) %*% ybar)
```

```
  theta <- mvtnorm::rmvnorm(1, mu.n, Lambda.n)
```

```
  Sn <- S0 + (t(azdiabetes_N) - c(theta)) %*% t( t(azdiabetes_N) - c(theta))
```

```
  Sigma <- solve( rWishart(1, nu0 + n, solve(Sn))[,1])
```

```
  THETA_n <- rbind(THETA_n, theta)
```

```
  SIGMA_n <- rbind(SIGMA_n, c(Sigma))
```

```
}
```

```
all_theta <- rbind(data.frame(THETA_d) %>% mutate(diabetes = "Yes"), data.frame(THETA_n) %>% mutate(diabetes = "No"))
colnames(all_theta) <- colnames(azdiabetes)
```

```
a1 <- all_theta %>% ggplot() + geom_density(aes(x=npreg, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
a2 <- all_theta %>% ggplot() + geom_density(aes(x=glu, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
a3 <- all_theta %>% ggplot() + geom_density(aes(x=bp, color = diabetes, fill = diabetes), alpha = 0.5)
```

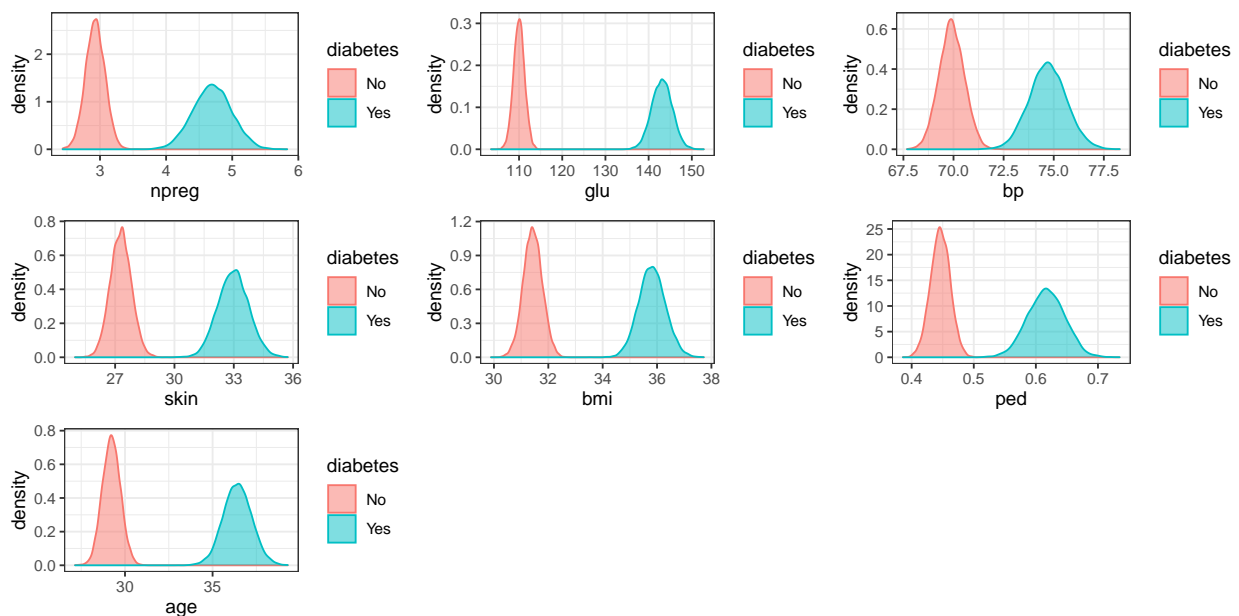
```
a4 <- all_theta %>% ggplot() + geom_density(aes(x=skin, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
a5 <- all_theta %>% ggplot() + geom_density(aes(x=bmi, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
a6 <- all_theta %>% ggplot() + geom_density(aes(x=ped, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
a7 <- all_theta %>% ggplot() + geom_density(aes(x=age, color = diabetes, fill = diabetes), alpha = 0.5)
```

```
grid.arrange(a1, a2, a3, a4, a5, a6, a7, nrow = 3)
```



We can see clear differences among all variables between two groups.

```

prob<-list()
for (i in 1:7)
{prob[[i]] <- mean(THETA_d[,i]>THETA_n[,i])}
unlist(prob)

```

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

$Pr(\theta_{d,j} > \theta_{n,j})$ is 1 for all seven j .

(b) Since we have 7 variables, each covaraince matrix should be a 7 by 7 matrix with 49 entries.

The plot suggests that for most of 49 entries, the one from Σ_d is positively correlated, or very similiar to the one from Σ_n , which is why we could see them roughly on the line with a slope of 1.

```
plot(x = colMeans(SIGMA_d), y = colMeans(SIGMA_n))
```

