S626

HW4

John Koo

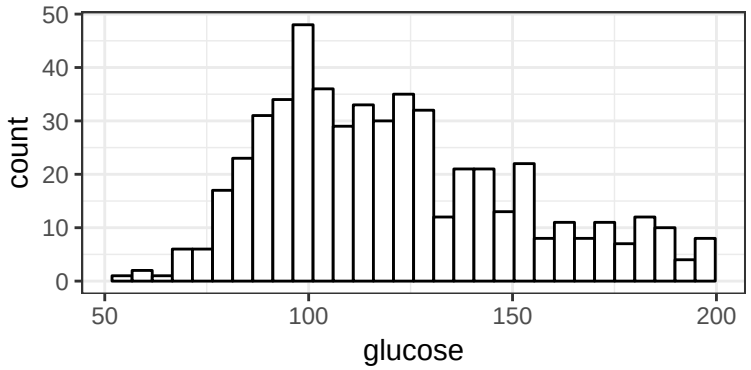
```
library(ggplot2)
import::from(magrittr, `%>%`)

theme_set(theme_bw())
set.seed(626)
doMC::registerDoMC(8)
```

6.2

```
glucose <-
  readLines('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/glucose.dat') %>%
  as.numeric()
```

a

```
ggplot() +
  geom_histogram(aes(x = glucose), colour = 'black', fill = 'white')
```



Although the empirical distribution looks approximately unimodal, it has a longer tail to the right.

b

We are given:

- $Y_i \mid x_i, \sim \mathcal{N}(\theta_{x_i}, \sigma^2_{x_i})$
- $X_i \mid p \sim Bernoulli(p)$
- $p \sim Beta(a, b)$
- $\theta_i \sim \mathcal{N}(\mu_0, \tau_0^2)$

- $1/\sigma_i^2 \sim Gamma(\nu_0/2, \nu_0\sigma_0^2/2)$

Then we can say:

- $p(y|\cdots) = \prod_i p(y_i|x_i, \cdots)$
  $= \prod_{i:x_i=0} p(y_i|\cdots) \prod_{i:x_i=1} p(y_i|\cdots)$
  $= \prod_i^n \mathcal{N}(y_i|\theta_1, \sigma_1^2)^{x_i} \mathcal{N}(y_i|\theta_2, \sigma_2^2)^{1-x_i}$
- X_i is still either 1 or 2, so $X_i \sim Bernoulli(\text{something})$
  $p(x_i|y_i, \cdots) = \frac{p(y_i|x_i, \cdots)p(x_i|\cdots)}{p(y_i|\cdots)}$
  $= \frac{\mathcal{N}(y_i|\theta_{x_i}, \sigma_{x_i}^2)p^{x_i}(1-p)^{1-x_i}}{p\mathcal{N}(y_i|\theta_1, \sigma_1^2)+(1-p)\mathcal{N}(y_i|\theta_2, \sigma_2^2)}$
  $\implies X_i \mid \cdots \sim Bernoulli\left(\frac{p\mathcal{N}(y_i|\theta_1, \sigma_1^2)}{p\mathcal{N}(y_i|\theta_1, \sigma_1^2)+(1-p)\mathcal{N}(y_i|\theta_2, \sigma_2^2)}\right)$
- $p(p|x, \cdots) \propto p(x|p)p(p)$
  $\propto p^{a-1}(1-p)^{b-1}p^{\sum_{x_i=1} x_i}(1-p)^{\sum_{x_i=2} x_i} \propto p^{a+n_1-1}(1-p)^{b+n_2-1}$, where n_1 is the number of times $x_i = 1$ and n_2 is the number of times $x_i = 2$
  $\implies p \mid \cdots \sim Beta(a + n_1, b + n_2)$
- $p(\theta_1|x, y, \cdots) \propto p(\theta_1)p(y|x, \theta_1, \cdots)$,
  $\propto p(\theta_1) \prod_i^n \mathcal{N}(y_i|\theta_1, \sigma_1^2)^{x_i}$
  and at this point this is just the posterior for the normal, and we can use our notes ...
  $\implies \theta_1 \mid \cdots \sim \mathcal{N}\left(\frac{n_1\bar{y}_1/\sigma_1^2+\mu_0/\tau_0^2}{n_1/\sigma_1^2+1/\tau_0^2}, (n_1/\sigma_1^2 + 1/\tau_0^2)^{-1}\right)$
  where $\bar{y}_1$ is the sample mean of y_i's that correspond to $x_i = 1$
- similarly, $\theta_2 \mid \cdots \sim \mathcal{N}\left(\frac{n_2\bar{y}_2/\sigma_2^2+\mu_0/\tau_0^2}{n_2/\sigma_2^2+1/\tau_0^2}, (n_2/\sigma_2^2 + 1/\tau_0^2)^{-1}\right)$
- similarly, $1/\sigma_1^2 \mid \cdots \sim Gamma(\frac{n_1+\nu_0}{2}, \frac{\sum_{i:x_i=1}(y_i-\theta_1)^2+\nu_0\sigma_0^2}{2})$
- similarly, $1/\sigma_2^2 \mid \cdots \sim Gamma(\frac{n_2+\nu_0}{2}, \frac{\sum_{i:x_i=2}(y_i-\theta_2)^2+\nu_0\sigma_0^2}{2})$

c

```
# iterations
iter <- 1e4

# priors
a <- b <- 1
mu0 <- 120
tau0 <- sqrt(200)
sigma0 <- sqrt(1000)
nu0 <- 10

# parameters
n <- length(glucose)

# initial values
# based on the data using kmeans
clusters <- kmeans(glucose, 2)$cluster
p <- mean(clusters == 1)
theta1 <- mean(glucose[clusters == 1])
theta2 <- mean(glucose[clusters == 2])
```

```r
sigma1 <- sd(glucose[clusters == 1])
sigma2 <- sd(glucose[clusters == 2])

# preallocate
X <- matrix(NA, nrow = iter, ncol = n)
P <- rep(NA, iter)
theta <- matrix(NA, nrow = iter, ncol = 2)
sigma <- matrix(NA, nrow = iter, ncol = 2)
N <- matrix(NA, nrow = iter, ncol = 2)
Y.pred <- rep(NA, iter)

# mcmc
for (i in seq(iter)) {
  # draw x
  x <- rbinom(
    n,
    1,
    p * dnorm(glucose, theta1, sigma1) /
      (p * dnorm(glucose, theta1, sigma1) +
         (1 - p) * dnorm(glucose, theta2, sigma2))
  )
  n1 <- sum(x)
  n2 <- n - n1

  # draw p
  p <- rbeta(1, a + n1, b + n2)

  # draw thetas
  theta1 <- rnorm(
    1,
    (sum(glucose[x == 1]) / sigma1 ** 2 + mu0 / tau0 ** 2) /
      (n1 / sigma1 ** 2 + tau0 ** -2),
    (n1 / sigma1 ** 2 + tau0 ** -2) ** -.5
  )
  theta2 <- rnorm(
    1,
    (sum(glucose[x == 0]) / sigma2 ** 2 + mu0 / tau0 ** 2) /
      (n2 / sigma2 ** 2 + tau0 ** -2),
    (n2 / sigma2 ** 2 + tau0 ** -2) ** -.5
  )

  # draw sigmas
  sigma1 <- rgamma(
    n = 1,
    shape = (n1 + nu0) / 2,
    rate = (sum((glucose[x == 1] - theta1) ** 2) + nu0 * sigma0 ** 2) /  2
  ) ** -.5
  sigma2 <- rgamma(
    n = 1,
    shape = (n2 + nu0) / 2,
    rate = (sum((glucose[x == 0] - theta2) ** 2) + nu0 * sigma0 ** 2) / 2
  ) ** -.5
```

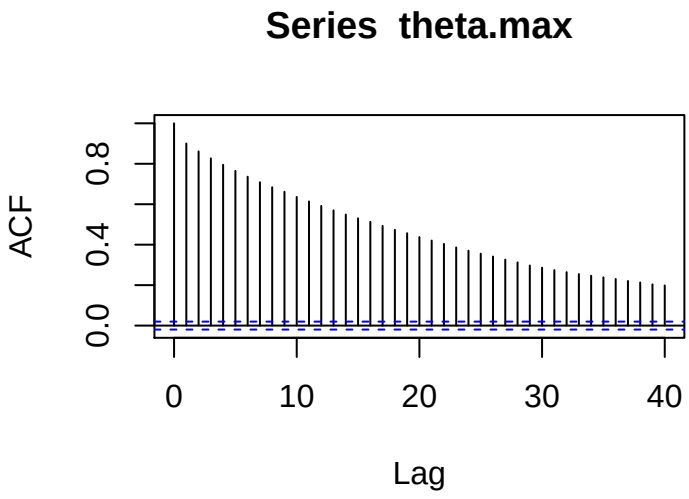
```
  # draw posterior pred
  y.pred <- ifelse(runif(1) < p,
                   rnorm(1, theta1, sigma1),
                   rnorm(1, theta2, sigma2))

  # store
  X[i, ] <- x
  P[i] <- p
  theta[i, ] <- c(theta1, theta2)
  sigma[i, ] <- c(sigma1, sigma2)
  N[i, ] <- c(n1, n2)
  Y.pred[i] <- y.pred
}

theta.max <- apply(theta, 1, max)
theta.min <- apply(theta, 1, min)

# diagnostics
coda::effectiveSize(theta.max)
```

```
    var1
215.8566
```

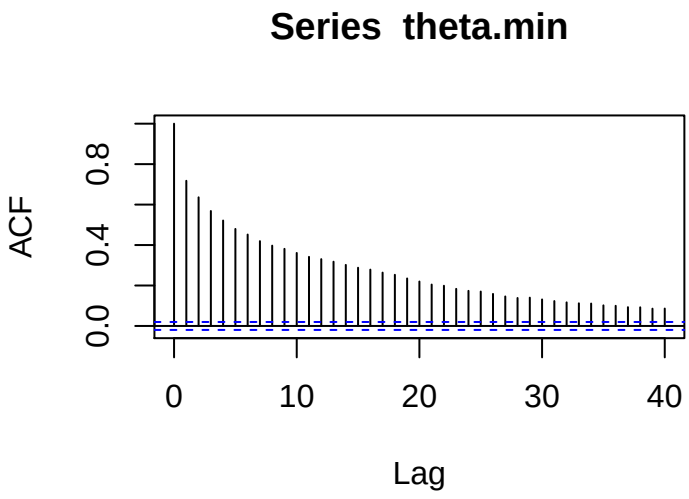
```
coda::effectiveSize(theta.min)
```

```
    var1
419.7371
```

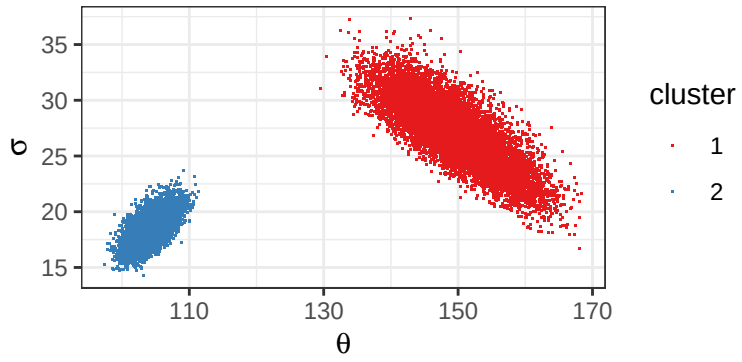
```
acf(theta.max)
```




```
acf(theta.min)
```

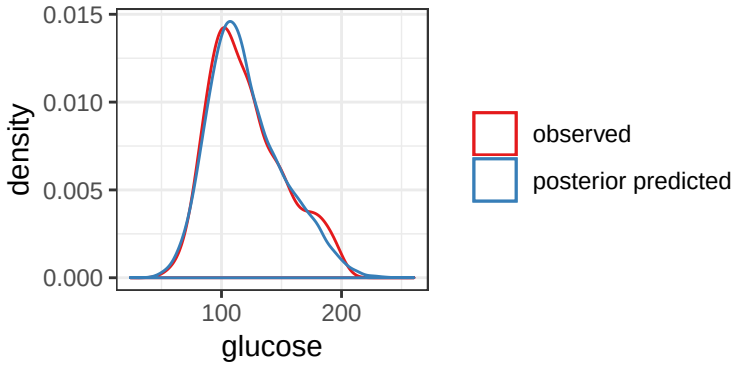



```
# posterior plot
ggplot() +
  geom_point(aes(x = theta[, 1], y = sigma[, 1], colour = '1'),
             shape = '.') +
  geom_point(aes(x = theta[, 2], y = sigma[, 2], colour = '2'),
             shape = '.') +
  labs(colour = 'cluster', x = expression(theta), y = expression(sigma)) +
  scale_colour_brewer(palette = 'Set1')
```



d

```
ggplot() +
  geom_density(aes(x = glucose, colour = 'observed')) +
  geom_density(aes(x = Y.pred, colour = 'posterior predicted')) +
  labs(colour = NULL) +
  scale_colour_brewer(palette = 'Set1')
```



The posterior predictive distribution matches the data very closely. We can do a Kolmogorov-Smirnov test to see if there is a significant difference between the two distributions.

```
ks.test(glucose, Y.pred)
```

```
    Two-sample Kolmogorov-Smirnov test

data:  glucose and Y.pred
D = 0.03199, p-value = 0.6794
alternative hypothesis: two-sided
```

7.3

```
bluecrab.df <-
  read.table('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/bluecrab.dat') %>%
  magrittr::set_colnames(c('depth', 'width'))

orangecrab.df <-
  read.table(
    'http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/orangecrab.dat') %>%
  magrittr::set_colnames(c('depth', 'width'))

crab.list <- list('blue' = bluecrab.df,
                  'orange' = orangecrab.df)
```

a

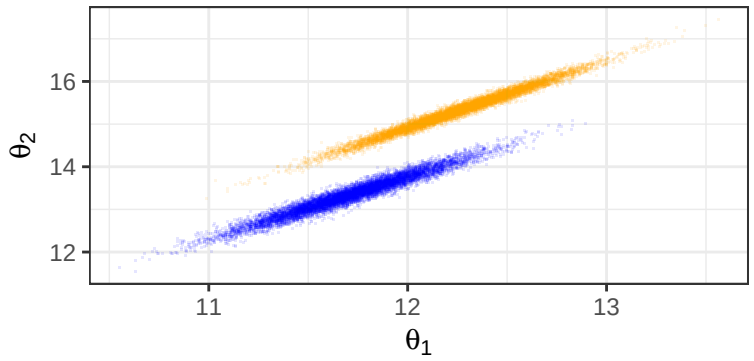
```
# global priors
nu0 <- 4

# samples to obtain
iter <- 1e4

# for each data frame ...
mcmc.out <- plyr::llply(crab.list, function(crab.df) {
  # local priors
  mu0 <- colMeans(crab.df)
```

```
  Lambda0 <- S0 <- cov(crab.df)
  Lambda0.inv <- solve(Lambda0)

  # starting values at sample statistics
  theta <- mu0
  Sigma <- S0

  # other sample statistics
  ybar <- mu0
  n <- nrow(crab.df)

  # preallocate
  theta.out <- matrix(NA, nrow = iter, ncol = 2)
  Sigma.out <- array(NA, c(iter, 2, 2))

  # mcmc
  for (i in seq(iter)) {
    # draw theta
    theta <- mvtnorm::rmvnorm(
      1,
      solve(Lambda0.inv + n * solve(Sigma)) %*%
        (n * solve(Sigma) %*% ybar + Lambda0.inv %*% mu0),
      solve(Lambda0.inv + n * solve(Sigma))
    )

    # draw Sigma
    Sigma <- solve(rWishart(
      1,
      n + nu0,
      solve(S0 + apply(crab.df, 1, function(x) x - theta) %>% {. %*% t(.)})
    )[, , 1])

    # store
    theta.out[i, ] <- theta
    Sigma.out[i, , ] <- Sigma
  }
  list(theta = theta.out, Sigma = Sigma.out)
}, .parallel = TRUE)
```

b

```
ggplot() +
  geom_point(aes(x = mcmc.out$blue$theta[, 1],
                 y = mcmc.out$blue$theta[, 2],
                 colour = 'blue'),
             colour = 'blue',
             pch = '.',
             alpha = .1) +
  geom_point(aes(x = mcmc.out$orange$theta[, 1],
                 y = mcmc.out$orange$theta[, 2],
                 colour = 'orange'),
             colour = 'orange',
```

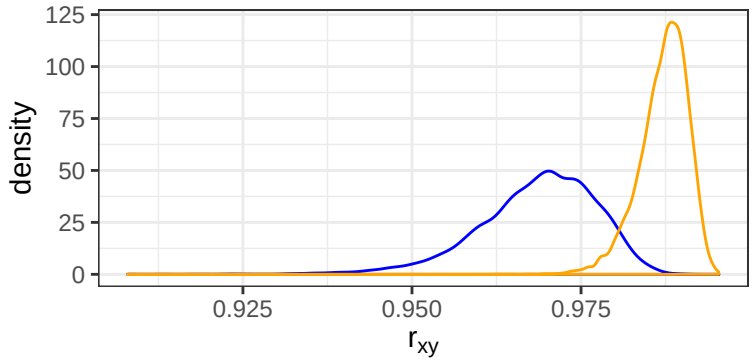
```
            pch = '.',
            alpha = .1) +
  labs(x = expression(theta[1]),
       y = expression(theta[2]))
```



It appears that orange crabs are larger than blue crabs, both in terms of rear width and body depth.

c

```
cor.df <- plyr::ldply(mcmc.out, function(color) {
  data.frame(sample.cor = apply(color$Sigma, 1, function(x) cov2cor(x)[1, 2]))
}, .parallel = TRUE) %>%
  dplyr::transmute(type = rep(c('blue', 'orange'), each = iter),
                   sample.cor)

ggplot(cor.df) +
  geom_density(aes(x = sample.cor, colour = type)) +
  labs(x = expression(r[xy])) +
  scale_colour_manual(values = c('blue', 'orange')) +
  theme(legend.position = 'none')
```




```
cor.df %>%
  dplyr::mutate(i = rep(seq(iter), 2)) %>%
  tidyr::spread(type, sample.cor) %>%
  dplyr::transmute(blue < orange) %>%
  colMeans()
```

```
blue < orange
```

```
0.9912
```

The body depth and rear width are much more strongly correlated for the orange crabs compared to the blue crabs.

7.4

```
agehw.df <-
  readr::read_delim('http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/agehw.dat',
                    delim = ' ')
```

a

We will make a guess without looking at the data but based on external sources. The median age of men and women in the US is 35.8 and 38.5 years respectively[1]. Typically, husbands are older than wives, so this prior is rather suspect.

The mean first marriage age in the US is 28.2[2]. I will add/subtract 5 years and assume a uniform distribution for the distribution of ages. Using the "half age + 7" rule[3] and assuming a uniform distribution again, we can generate the ages of spouses:

```
mean.age <- 28.2
size <- 5
samples <- 1e3

plyr::ldply(seq(samples), function(i) {
  age <- runif(1, mean.age - size, mean.age + size)
  age.partner <- runif(1, age / 2 + 7, 2 * (age - 7))
  dplyr::data_frame(age = age, age.partner = age.partner)
}, .parallel = TRUE) %>%
  dplyr::summarise(cor(age, age.partner))
```

```
  cor(age, age.partner)
1             0.5263391
```

To simplify, I will say the correlation is 0.5.

For the variances, I couldn't find any information from a quick search, so I will just assume $10^2 = 100$ for either gender.

Since these are bad priors, $\nu_0 = p + 1 = 3$.

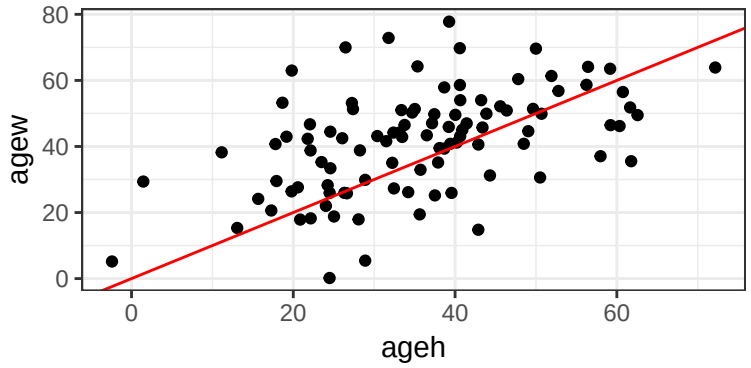
b

```
# priors
nu0 <- 3
mu0 <- c(35.8, 38.5)
Lambda0 <- S0 <- 100 * rbind(c(1, .5), c(.5, 1))
Lambda0.inv <- solve(Lambda0)
```

[1] https://www.census.gov/prod/cen2010/briefs/c2010br-03.pdf
[2] https://en.wikipedia.org/wiki/List_of_countries_by_age_at_first_marriage
[3] https://xkcd.com/314/

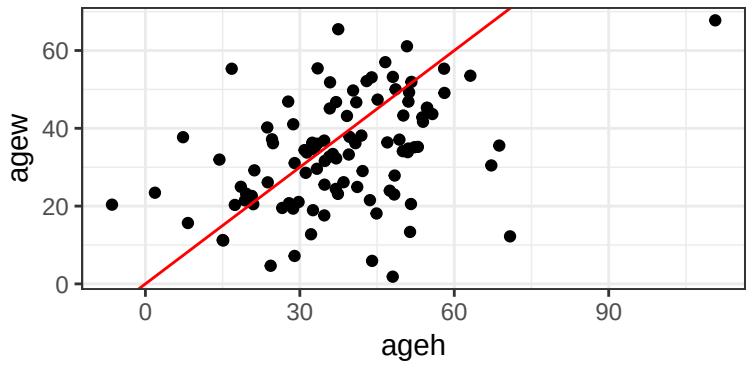
```r
plot.prior.pred <- function(nu0, mu0, Lambda0, S0, n = 100) {
  # generate a prior predictive dataset based on priors
  # then plot

  plyr::ldply(seq(n), function(i) {
    mu <- mvtnorm::rmvnorm(1, mu0, Lambda0)
    Sigma <- solve(rWishart(1, nu0, solve(S0))[, , 1])
    mvtnorm::rmvnorm(1, mu, Sigma) %>%
      as.data.frame() %>%
      magrittr::set_colnames(c('ageh', 'agew'))
  }) %>%
    ggplot() +
    geom_point(aes(x = ageh, y = agew)) +
    geom_abline(colour = 'red')
}
```

```r
plot.prior.pred(nu0, mu0, Lambda0, S0)
```



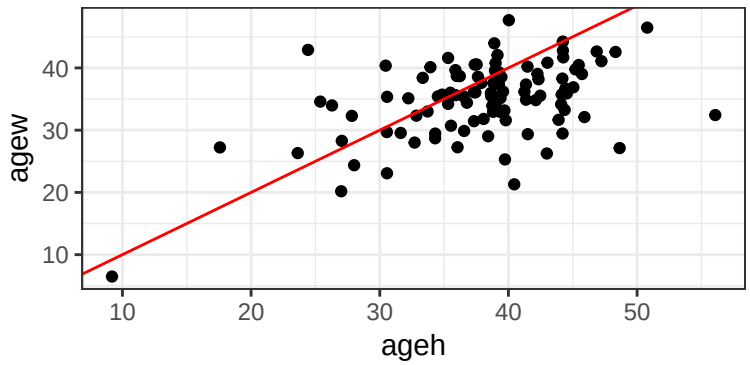
The main issue with our prior is that we know that husbands tend to be older than wives. So instead of using the median ages of men and women, we can look at the median ages for men and women to first get married, then add a few years until we approach the median age of people in the US.

```r
mu0 <- c(29.2, 27.1)
overall.mean <- 37.2
difference <- overall.mean - mean(mu0)
mu0 <- mu0 + difference

plot.prior.pred(nu0, mu0, Lambda0, S0)
```



The data are much more diffuse than we would like. For example, no one should be below around 20 or over around 90. We can try tightening this by adjusting Λ_0 and ν_0.

```
Lambda0 <- 10 * rbind(c(1, .5), c(.5, 1))
nu0 <- 5

plot.prior.pred(nu0, mu0, Lambda0, S0)
```

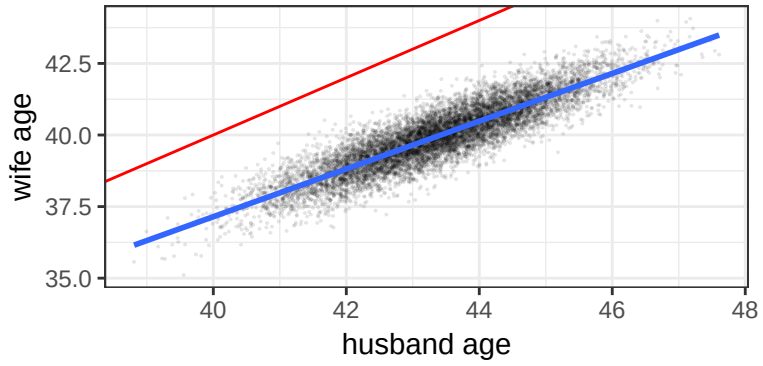


We still get some rather strange values, but those seem rare enough.

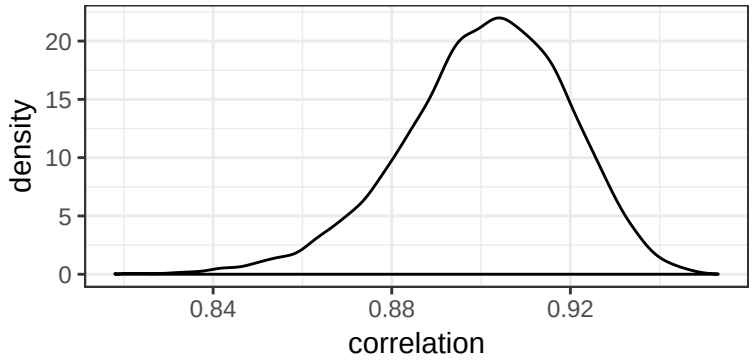
c

```
# precompute
Lambda0.inv <- solve(Lambda0)

# initial guesses at sample statistics
mu <- colMeans(agehw.df)
Sigma <- cov(agehw.df)

# sample statistics
ybar <- mu
n <- nrow(agehw.df)

# preallocate
mu.out <- matrix(NA, nrow = iter, ncol = 2)
Sigma.out <- array(NA, c(iter, 2, 2))
cor.out <- rep(NA, iter)

# mcmc
for (i in seq(iter)) {
  # precompute
  Phi <- solve(Sigma)

  # draw mu
  mu <- mvtnorm::rmvnorm(
    1,
    solve(Lambda0.inv + n * Phi) %*% (n * Phi %*% ybar + Lambda0.inv %*% mu0),
    solve(Lambda0.inv + n * Phi)
  )

  # draw Sigma
```

```
  Sigma <- solve(rWishart(
    1,
    n + nu0,
    solve(S0 + apply(agehw.df, 1, function(x) x - mu) %>% {. %*% t(.)})
  )[, , 1])

  # store
  mu.out[i, ] <- mu
  Sigma.out[i, , ] <- Sigma
  cor.out[i] <- cov2cor(Sigma)[1, 2]
}

# plot of thetas
mu.out %>%
  as.data.frame() %>%
  ggplot() +
  geom_point(aes(x = V1, y = V2), alpha = .1, size = .01) +
  labs(x = 'husband age', y = 'wife age') +
  geom_abline(colour = 'red') +
  stat_smooth(aes(x = V1, y = V2), method = 'lm')
```




```
# plot of correlation
ggplot() +
  geom_density(aes(x = cor.out)) +
  labs(x = 'correlation')
```




```
# intervals
alpha <- .05
mu.out %>%
```

```
  as.data.frame() %>%
  magrittr::set_colnames(c('husband', 'wife')) %>%
  plyr::llply(function(x) {
    quantile(x, c(alpha / 2, 1 - alpha / 2))
  })
```

```
$husband
    2.5%    97.5%
40.92144 45.78631

$wife
    2.5%    97.5%
37.67965 42.23352
```

```
quantile(cor.out, c(alpha / 2, 1 - alpha / 2))
```

```
     2.5%     97.5%
0.8610047 0.9330025
```