# 1. Exercise 7.1 (pg 238) Jeffreys' prior: For the multivariate normal

model, Jeffreys' rule for generating a prior distribution on $(\theta, \Sigma)$ gives $p_J(\theta, \Sigma) \propto |\Sigma|^{-(p+2)/2}$.

## a) Explain why the function $p_J$ cannot actually be a probability density for $(\theta, \Sigma)$.

Since the density is uniform with respect to $\boldsymbol{\theta}$, the integral over the support of this function is infinite and cannot be 1.

## b) Let $p_J(\theta, \Sigma|y_1, ..., y_n)$ be the probability density that is proportional

to $p_J(\theta, \Sigma) \times p(y_1, ..., y_n|\theta, \Sigma)$.Obtain the form of $p_J(\theta, \Sigma|y_1, ..., y_n)$, $p_J(\theta|\Sigma, y_1, ..., y_n)$ and $p_J(\Sigma|y_1, ..., y_n)$.

$$\begin{aligned}
p_J(\boldsymbol{\theta}, \Sigma \mid \boldsymbol{y}_{1:n}) &\propto p(\boldsymbol{\theta}, \Sigma) \times p(\boldsymbol{y}_{1:n} \mid \boldsymbol{\theta}, \Sigma) \\
&\propto \left(|\Sigma|^{-\frac{p+2}{2}}\right) \times \left(|\Sigma|^{-\frac{n}{2}} \exp\left[-\frac{1}{2}\text{tr}(\mathbf{S}_\theta \Sigma^{-1})\right]\right) \\
&\propto |\Sigma|^{-\frac{n+p+2}{2}} \exp\left[-\frac{1}{2}\text{tr}(\mathbf{S}_\theta \Sigma^{-1})\right]
\end{aligned}$$

To obtain the full conditionals of a parameter, we treat the other parameters as constant, so

$$\begin{aligned}
p_J(\boldsymbol{\theta} \mid \Sigma, \boldsymbol{y}_{1:n}) &\propto \exp\left[-\frac{1}{2}\text{tr}(\mathbf{S}_\theta \Sigma^{-1})\right] \\
&= \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \theta)'\Sigma^{-1}(\boldsymbol{y}_i - \theta)\right] \\
&= \exp\left[-\frac{n}{2}(\bar{\boldsymbol{y}} - \theta)'\Sigma^{-1}(\bar{\boldsymbol{y}} - \theta)\right] \\
\boldsymbol{\theta} \mid \Sigma, \boldsymbol{y}_{1:n} &\sim \text{Normal}(\bar{\boldsymbol{y}}, \Sigma/n) \\
p_J(\Sigma \mid \boldsymbol{\theta}, \boldsymbol{y}_{1:n}) &\propto |\Sigma|^{-\frac{n+p+2}{2}} \exp\left[-\frac{1}{2}\text{tr}(\mathbf{S}_\theta \Sigma^{-1})\right] \\
\Sigma \mid \boldsymbol{\theta}, \boldsymbol{y}_{1:n} &\sim \text{Inverse-Wishart}\left(n+1, \mathbf{S}_\theta^{-1}\right)
\end{aligned}$$

# 2. Exercise 7.2 (pg 238) Unit information prior

Letting $\Psi = \Sigma^{-1}$, show that a unit information prior for $(\theta, \Psi)$ is given by $\theta|\Psi \sim$ multivariate normal$(\bar{y}, \Psi^{-1})$ and $\Psi \sim$ Wishart$(p+1, S^{-1})$, where $S = \sum(y_i - \bar{y})(y_i - \bar{y})^T/n$. This can be done by mimicking the procedure outlined in Exercise 5.6 as follows:

## a) Reparameterize the multivariate normal model in terms of the precision matrix

$\Psi = \Sigma^{-1}$. Write out the resulting log likelihood, and find a probability density $p_U(\theta, \Psi) = p_U(\theta|\Psi)p_U(\Psi)$ such that $\log p(\theta, \Psi) = l(\theta, \Psi|\mathbf{Y})/n + c$, where c does not depend on $\theta$ or $\Psi$.

Hint: Write $(y_i - \theta)$ as $(y_i - \bar{y} + \bar{y} - \theta)$, and note that $\sum a_i^T \mathbf{B} a_i$ can be written as $\mathrm{tr}(AB)$, where $\mathbf{A} = \sum a_i a_i^T$ .

$$\log p(\theta, \Psi) = \frac{1}{n} l(\theta, \Psi|\mathbf{Y}) + c = \ln p_U(\theta|\Psi) + \ln p_U(\Psi)$$

$$l(\theta, \Psi|\mathbf{Y}) = \ln[\prod_{i=1}^{n} p(y_i|\theta, \Psi)] = \frac{-np}{2}\ln(2\pi) - \frac{n}{2}\ln(|\Psi^{-1}|) - \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta)^T \Psi (y_i - \theta)$$

$$= \frac{-np}{2}\ln(2\pi) - \frac{n}{2}\ln(|\Psi^{-1}|) - \frac{1}{2}\mathrm{tr}\left[\sum_{i=1}^{n}(y_i - \bar{y} + \bar{y} - \theta)^T \Psi (y_i - \bar{y} + \bar{y} - \theta)\right]$$

$$= \frac{-np}{2}\ln(2\pi) - \frac{n}{2}\ln(|\Psi^{-1}|) - \frac{1}{2}\mathrm{tr}\{\Psi[\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T}_{nS} + (\bar{y} - \theta)\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^T}_{0} + \underbrace{\sum_{i=1}^{n}(y_i - \bar{y})}_{0}(\bar{y} - \theta)^T + n(\bar{y} - \theta)(\bar{y} - \theta)^T]\}$$

$$= \underbrace{-\frac{1}{2}\mathrm{tr}\left[\Psi n S\right]}_{n \ln p_U(\Psi)} - \underbrace{\frac{n}{2}\ln(|\Psi^{-1}|) - \frac{n}{2}\mathrm{tr}\left[\Psi(\bar{y} - \theta)(\bar{y} - \theta)^T\right]}_{n \ln p_U(\theta|\Psi)} + \frac{-np}{2}\ln(2\pi)$$

$$\ln p_U(\Psi) = -\frac{1}{2n}\mathrm{tr}\left[\Psi \sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T\right] = -\frac{1}{2}\mathrm{tr}\left[S\Psi\right]$$

$$p_U(\Psi) \propto |\Psi|^{\frac{p+1-p-1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[S\Psi\right]\right\}$$

$$\Psi \sim \mathrm{Wishart}(p+1, S^{-1})$$

$$\ln p_U(\theta|\Psi) = \frac{1}{2}\ln(|\Psi^{-1}|) - \frac{1}{2}\mathrm{tr}\left[\Psi(\bar{y} - \theta)(\bar{y} - \theta)^T\right]$$

$$p_U(\theta|\Psi) = |\Psi^{-1}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[\Psi(\bar{y} - \theta)(\bar{y} - \theta)^T\right]\right\} = |\Psi^{-1}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\bar{y} - \theta)\Psi(\bar{y} - \theta)^T\right\}$$

$$\theta|\Psi \sim \text{multivariate normal}(\bar{y}, \Psi^{-1})$$

## b) Let $p_U(\Sigma)$ be the inverse-Wishart density induced by $p_U(\Psi)$.

Obtain a density $p_U(\theta, \Sigma|y_1, ..., y_n) \propto\ p_U(\theta|\Sigma)p_U(\Sigma)p(y_1, ..., y_n|\theta, \Sigma)$. Can this be interpreted as a posterior distribution for $\theta$ and $\Sigma$?

$$p_U(\Sigma) \propto |\Sigma|^{-\frac{p+1+1+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[S\Sigma^{-1}\right]\right\}$$

$$p_U(\theta|\Sigma) \propto |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\bar{y} - \theta)\Sigma^{-1}(\bar{y} - \theta)^T\right\}$$

$$p(y_{1:n}|\theta, \Sigma) \propto |\Sigma|^{-\frac{n}{2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta)^T \Sigma^{-1}(y_i - \theta)\right]$$

$$p_U(\theta, \Sigma|y_{1:n}) \propto\ p_U(\theta|\Sigma)p_U(\Sigma)p(y_{1:n}|\theta, \Sigma)$$

$$\propto |\Sigma|^{-\frac{1}{2}-\frac{p+3}{2}-\frac{n}{2}} \exp\left[-\frac{1}{2}(\bar{y}-\theta)\Sigma^{-1}(\bar{y}-\theta)^T - \frac{1}{2}\mathrm{tr}\left[S\Sigma^{-1}\right] - \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta)\right]$$

$$\propto |\Sigma|^{-\frac{1}{2}-\frac{n+p+3}{2}} \exp[-\frac{1}{2}(\bar{y}-\theta)\Sigma^{-1}(\bar{y}-\theta)^T - \frac{1}{2}\mathrm{tr}\left[S\Sigma^{-1}\right] - \frac{1}{2}\sum_{i=1}^{n}(y_i-\bar{y})\Sigma^{-1}(y_i-\bar{y})^T$$

$$+\frac{1}{2}\sum_{i=1}^{n}(y_i-\bar{y})\Sigma^{-1}(y_i-\bar{y})^T - \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta)^T\Sigma^{-1}(y_i-\theta)]$$

$$\propto |\Sigma|^{-\frac{1}{2}-\frac{n+p+3}{2}} \exp\left\{-\frac{1}{2}\left[(\bar{y}-\theta)\frac{\Sigma^{-1}}{n}(\bar{y}-\theta)^T + \sum_{i=1}^{n}(\bar{y}-\theta)^T\Sigma^{-1}(\bar{y}-\theta)\right] - \frac{1}{2}\mathrm{tr}\left[S\Sigma^{-1}\right] - \frac{n}{2}\mathrm{tr}\left[S\Sigma^{-1}\right]\right\}$$

$$\propto |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[(\bar{y}-\theta)(n+1)\Sigma^{-1}(\bar{y}-\theta)^T\right]\right\} |\Sigma|^{-\frac{n+p+3}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[(n+1)S\Sigma^{-1}\right]\right\}$$

$$\propto |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[(\bar{y}-\theta)(n+1)\Sigma^{-1}(\bar{y}-\theta)^T\right]\right\} |\Sigma|^{-\frac{n+p+3}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\left[(n+1)S\Sigma^{-1}\right]\right\}$$

$$\theta,\Sigma|y_{1:n} \sim \mathrm{MVN}(\theta|\bar{y},\frac{\Sigma}{n+1})\cdot\text{Inverse-Wishart}(\Sigma|n+2,\frac{1}{(n+1)S})$$

density induced by $p_U(\Psi)$. Obtain a density $p_U(\theta,\Sigma|y_1,...,y_n) \propto p_U(\theta|\Sigma)p_U(\Sigma)p(y_1,...,y_n|\theta,\Sigma)$. Can this be interpreted as a posterior distribution for $\theta$ and $\Sigma$

## 3. Exercise 7.4 (pg 239) Marriage data

The file agehw.dat contains data on the ages of 100 married couples sampled from the U.S. population.

```r
#Store the agehw.dat files in the same folder as this Rmd file
Y.marr <- read.table("agehw.dat",sep=" ",header=T)
```

### a) Before you look at the data, use your own knowledge to formulate a semiconjugate prior distribution for

$\theta=(\theta_h,\theta_w)^T$ and $\Sigma$,where $\theta_h$,$\theta_w$ are mean husband and wife ages, and $\Sigma$ is the covariance matrix.

Assume the mean value is 40. The 95% range of ages is [20,60]. Variance is $10^2=100$. Correlation is 0.9, $\sigma_{hw}=0.9\times 100=90$. Set

$$\mathbf{S}_0^{-1}=\Lambda_0=\begin{bmatrix}100 & 90\\ 90 & 100\end{bmatrix} \qquad \nu_0=p+2=4$$

```r
Y = Y.marr
p = ncol(Y.marr)
n = nrow(Y.marr)
ybar = colMeans(Y.marr)
mu0 = rep(40, p)
lambda0 = s0 = rbind(c(100,90), c(90,100))
nu0 = p + 2
```

## b) Generate a prior predictive dataset of size n = 100,

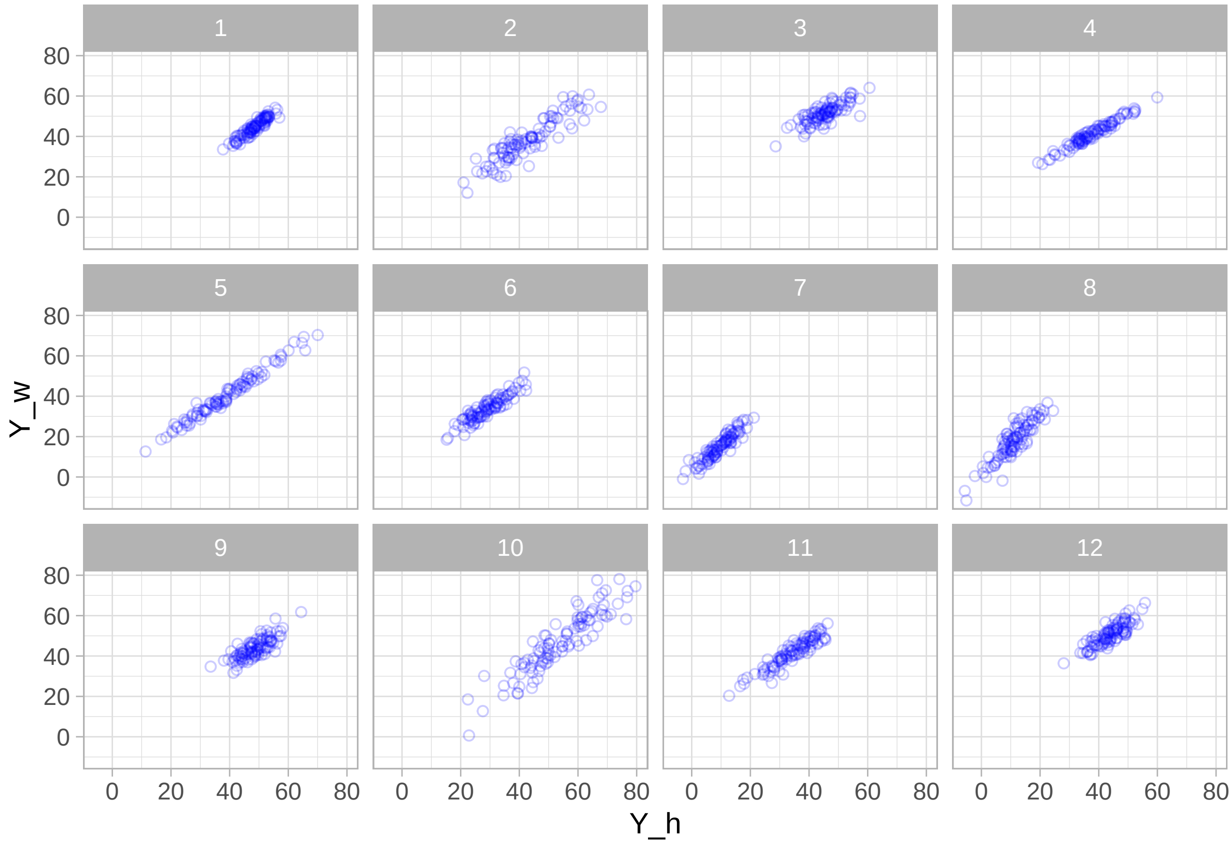
by sampling $(\theta, \Sigma)$ from your prior distribution and then simulating $Y_1, ..., Y_n \sim$ i.i.d. multivariate normal$(\theta, \Sigma)$. Generate several such datasets, make bivariate scatterplots for each dataset, and make sure they roughly represent your prior beliefs about what such a dataset would actually look like. If your prior predictive datasets do not conform to your beliefs, go back to part a) and formulate a new prior. Report the prior that you eventually decide upon, and provide scatterplots for at least three prior predictive datasets.

The wording of the question is interesting - I assume I'm supposed to sample a fixed $\boldsymbol{\theta}, \Sigma$ and from there sample 100 points all with the same parameters. If I were to do this myself, I feel like I would sample a new data point for each sample of $\boldsymbol{\theta}, \Sigma$...

In fact, because of that wording, I originally set $\nu_0 = p + 2 = 4$ to loosely center my prior. But given that this variance will often produce uncorrelated prior predictive datasets, I'm increasing $\nu_0$ a bit...

After increasing $\nu_0$, I'm fairly comfortable with what these posterior predictive datasets look like.

```r
# N = 100;S = 12
Y_preds = lapply(1:12, function(s) {
  # Sample THETA according to prior
  theta = mvrnorm(n = 1, mu0, lambda0)
  sigma = solve(rWishart(1, nu0, solve(s0))[, , 1])
  Y_s = mvrnorm(n = 100, theta, sigma)
  data.frame(Y_h = Y_s[, 1], Y_w = Y_s[, 2], dataset = s)
})
Y_comb = do.call(rbind, Y_preds)
ggplot(Y_comb, aes(x = Y_h, y = Y_w)) +geom_point(shape =1,alpha = 2/10,colour="blue") +facet_wrap(~ da
```



### c) Using your prior distribution and the 100 values in the dataset,

obtain an MCMC approximation to $p(\theta, \Sigma | y_1, ..., y_{100})$. Plot the joint posterior distribution of $\theta_h$ and $\theta_w$, and also the marginal posterior density of the correlation between $Y_h$ and $Y_w$, the ages of a husband and wife. Obtain 95% posterior confidence intervals for $\theta_h$, $\theta_w$ and the correlation coefficient.

```
S = 10000
mcmc = function(Y, mu0, lambda0, s0, nu0) {
  ybar = colMeans(Y); p = ncol(Y); n = nrow(Y)
  THETA = matrix(nrow = S, ncol = p)
  SIGMA = array(dim = c(p, p, S))
  sigma = cov(Y) # Start with sigma sample
  # Gibbs sampling
  for (s in 1:S) {
    # Update theta
    lambda_n = solve(solve(lambda0) + n * solve(sigma))
    mu_n = lambda_n %*% (solve(lambda0) %*% mu0 + n * solve(sigma) %*% ybar)
    theta = mvrnorm(n = 1, mu_n, lambda_n)
    # Update sigma
    resid = t(Y) - c(theta)
    s_theta = resid %*% t(resid)
    s_n = s0 + s_theta
    sigma = solve(rWishart(1, nu0 + n,solve(s_n))[, , 1])

    THETA[s, ] = theta
    SIGMA[, , s] = sigma
  }
  list(theta = THETA, sigma = SIGMA)
}
prior_mcmc = mcmc(Y.marr, mu0, lambda0, s0, nu0)
THETA = prior_mcmc$theta
SIGMA = prior_mcmc$sigma

print_quantiles = function(THETA, SIGMA) {
  print("Husband")
  print(quantile(THETA[, 1], probs = c(0.025, 0.5, 0.975))) # Husband
  print("Wife")
  print(quantile(THETA[, 2], probs = c(0.025, 0.5, 0.975))) # Wife
  cors = apply(SIGMA, MARGIN = 3, FUN = function(covmat) {
    covmat[1, 2] / (sqrt(covmat[1, 1] * covmat[2, 2]))
  })
  print("Correlation")
  print(quantile(cors, probs = c(0.025, 0.5, 0.975)))
}
print_quantiles(THETA, SIGMA)
```

```
## [1] "Husband"
##     2.5%      50%    97.5%
## 41.65563 44.32248 46.93694
## [1] "Wife"
##     2.5%      50%    97.5%
## 38.32931 40.84134 43.35535
## [1] "Correlation"
##      2.5%       50%     97.5%
## 0.8614125 0.9040568 0.9340423
```

## d) Obtain 95% posterior confidence intervals for $\theta_h$, $\theta_w$ and the correlation coefficient using the following prior distributions:

### i. Jeffreys' prior in Exercise 7.1;

```
THETA = matrix(nrow = S, ncol = p)
SIGMA = array(dim = c(p, p, S))
sigma = cov(Y)# Start with sigma sample
# Gibbs sampling
for (s in 1:S) {
  # Update theta
  theta = mvrnorm(n = 1, ybar, sigma/n)
  # Update sigma
  resid = t(Y) - c(theta)
  s_theta = resid %*% t(resid)
  sigma = solve(rWishart(1, n + 1, solve(s_theta))[, , 1])
  THETA[s, ] = theta
  SIGMA[, , s] = sigma
}
print_quantiles(THETA, SIGMA)
```

```
## [1] "Husband"
##     2.5%      50%    97.5%
## 41.71471 44.42065 47.08346
## [1] "Wife"
##     2.5%      50%    97.5%
## 38.34380 40.88508 43.41145
## [1] "Correlation"
##      2.5%       50%     97.5%
## 0.8605209 0.9039750 0.9345540
```

### iii. a "diffuse prior" with $\mu_0 = \mathbf{0}, \Lambda_0 = 10^5 \times \mathbf{I}, \mathbf{S_0} = 1000 \times \mathbf{I}$ and $\nu_0 = 3$.

```
mu0 = rep(0, p)
lambda0 = 10^5 * diag(p)
s0 = 1000 * diag(p)
nu0 = 3
diffuse_mcmc = mcmc(Y.marr, mu0, lambda0, s0, nu0)
print_quantiles(diffuse_mcmc$theta, diffuse_mcmc$sigma)
```

```
## [1] "Husband"
##     2.5%      50%    97.5%
## 41.70513 44.43630 47.13738
## [1] "Wife"
##     2.5%      50%    97.5%
## 38.31591 40.89315 43.47764
## [1] "Correlation"
##      2.5%       50%     97.5%
## 0.7932531 0.8552080 0.8994995
```

## e) Compare the confidence intervals from d) to those obtained in c).

Discusswhetherornotyouthinkthatyourpriorinformationishelpful in estimating $\theta$ and $\Sigma$, or if you think one of the alternatives in d) is preferable. What about if the sample size were much smaller, say $n = 25$?

- My prior

```
mu0 = rep(40, p)
lambda0 = s0 = rbind(c(100,90), c(90,100))
nu0 = p + 2
# nu0 = p + 2 + 10
prior_mcmc_short = mcmc(Y.marr[1:25,], mu0, lambda0, s0, nu0)
print_quantiles(prior_mcmc_short$theta, prior_mcmc_short$sigma)
```

```
## [1] "Husband"
##     2.5%      50%    97.5%
## 39.73300 44.81270 49.91035
## [1] "Wife"
##     2.5%      50%    97.5%
## 37.34750 42.57897 47.90849
## [1] "Correlation"
##      2.5%       50%     97.5%
## 0.8379177 0.9208696 0.9625000
```

- Diffuse prior

```
mu0 = rep(0, p)
lambda0 = 10^5 * diag(p)
s0 = 1000 * diag(p)
nu0 = 3
diffuse_mcmc_short = mcmc(Y.marr[1:25,], mu0, lambda0, s0, nu0)
print_quantiles(diffuse_mcmc_short$theta, diffuse_mcmc_short$sigma)
```

```
## [1] "Husband"
##     2.5%      50%    97.5%
## 39.30519 45.10721 50.96931
## [1] "Wife"
##     2.5%      50%    97.5%
## 36.74924 42.75979 48.88741
## [1] "Correlation"
##      2.5%       50%     97.5%
## 0.5395522 0.7605685 0.8827324
```

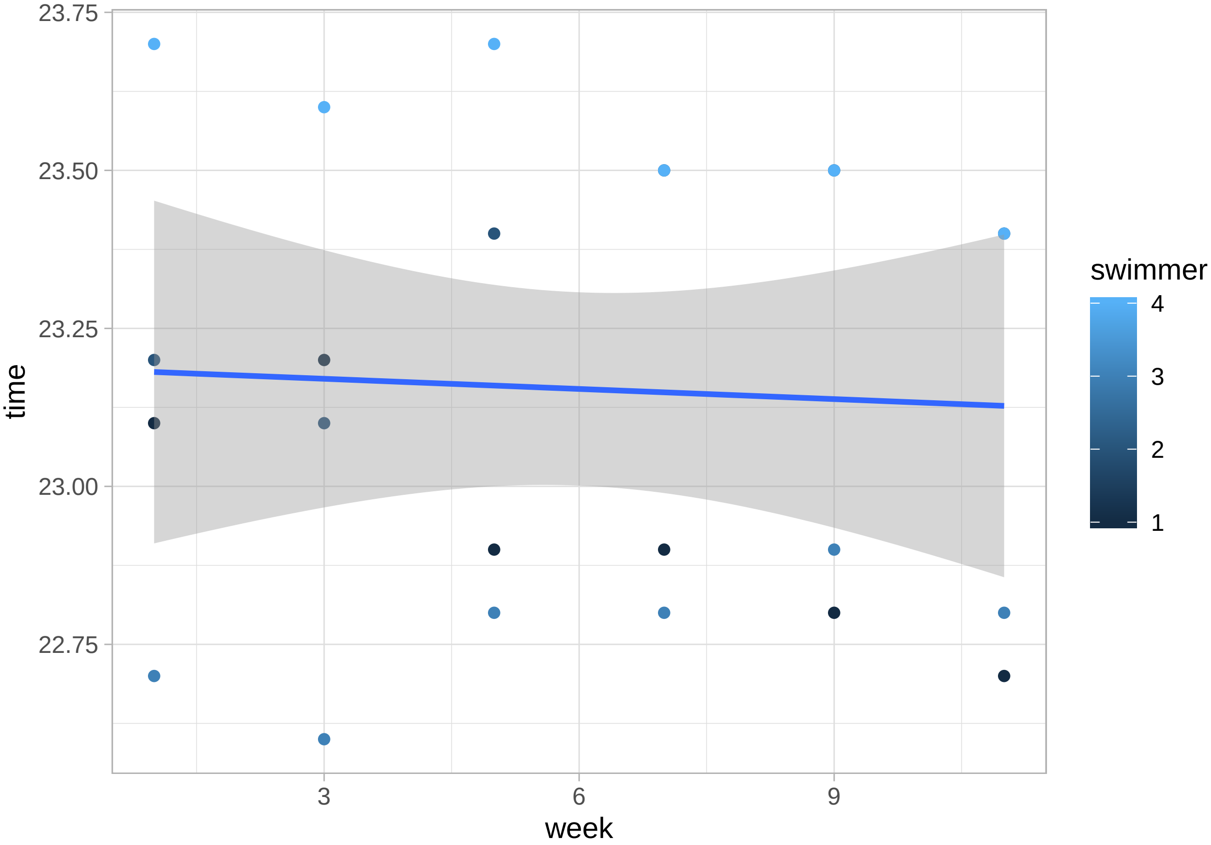
# 4. Exercise 9.1 (pg 242) Extrapolation

The file swim.dat contains data on the amount of time, in seconds, it takes each of four high school swimmers to swim 50 yards. Each swimmer has six times, taken on a biweekly basis.

## a) Perform the following data analysis for each swimmer separately:

i. Fit a linear regression model of swimming time as the response and week as the explanatory variable. To formulate your prior, use the information that competitive times for this age group generally range from 22 to 24 seconds.

```
model_swim <- lm(time~week,data.frame(Y.swim))
ggplot(data.frame(Y.swim),aes(x=week,y=time,colour=swimmer))+geom_point()+geom_smooth(method ='lm')+ther
```



ii. For each swimmer j, obtain a posterior predictive distribution for $Y_j^\star$ , their time if they were to swim two weeks from the last recorded time.

To specify our prior, we let the prior expectation of our $y$-intercept to be 23, and we let the prior expectation of the effect of training week by week to be 0, so $\boldsymbol{\beta} = (23, 0)^T$. We expect no covariance with the $\beta$ coefficients, but we do have uncertainty about our initial $\beta$ estimates. Specifically, to let 95% of our uncertainty of the $y$-intercept to fall in $[22, 24]$, we let $\Sigma_{0(1,1)} = 1/4$ (so that ± 2 standard deviations is ± 1). We also expect that training has a relatively mild effect on time, so we let $\Sigma_{0(2,2)} = 0.1$ which is just an arbitrarily chosen small variance. For our expectation of the variability of measurements, let's similarly set $\sigma_0^2 = 1/4$ and only lightly center this prior with $\nu_0 = 1$.

```
S = 5000
X = cbind(rep(1, 6), seq(1, 11, by = 2))
n = dim(X)[1]
p = dim(X)[2]
# Prior
beta0 = c(23, 0)
sigma0 = rbind(c(0.25, 0), c(0, 0.2))
nu0 = 1
s20 = 0.25
set.seed(1)
# run linear regression gibbs sampling and obtain a posterior for each swimmer
# predictive distribution
swim_pred = apply(swim, MARGIN = 1, function(y) {
  BETA = matrix(nrow = S, ncol = length(beta0)) # Store samples
  SIGMA = numeric(S)
  beta = c(23, 0) # Starting values
  s2 = 0.7^2
  for (s in 1:S) { # Gibbs sampling algorithm from 9.2.1
    V = solve(solve(sigma0) + (t(X) %*% X) / s2) # 1a) Compute V and m
    m = V %*% (solve(sigma0) %*% beta0 + (t(X) %*% y) / s2)
    beta = mvrnorm(1, m, V) # 1b) sample beta
    ssr=(t(y)%*%y)-(2*t(beta) %*% t(X)%*%y)+(t(beta)%*%t(X)%*%X%*%beta) # 2a) Compute SSR(beta) (from 9
    s2 = 1 / rgamma(1, (nu0 + n) / 2, (nu0 * s20 + ssr) / 2) # 2b) sample s2
    BETA[s, ] = beta
    SIGMA[s] = s2
  }
  xpred = c(1, 13) # sample posterior predictive + two weeks
  YPRED = rnorm(S, BETA %*% xpred, sqrt(SIGMA))
  YPRED
})
```

### b) The coach of the team has to decide which of the four swimmers will compete

in a swimming meet in two weeks. Using your predictive distributions, compute $Pr(Y_j^\star = \max\{Y_1^\star, ..., Y_4^\star\}|Y)$ for each swimmer j , and based on this make a recommendation to the coach.

```
fastest_times = apply(swim_pred, MARGIN = 1, FUN = which.min)
table(fastest_times) / length(fastest_times)
```

```
## fastest_times
##      1      2      3      4
## 0.6534 0.0134 0.3050 0.0282
```

In posterior predictive dataset, swimmer 1 is the fastest about 65% of the time by week 13, so we recommend that swimmer 1 race.

## 5. Exercise 9.3 (pg 243) Crime

The file crime.dat contains crime rates and data on 15 explanatory variables for 47 U.S. states, in which both the crime rates and the explanatory variables have been centered and scaled to have variance 1. A description of the variables can be obtained by typing library(MASS);?UScrime in R.

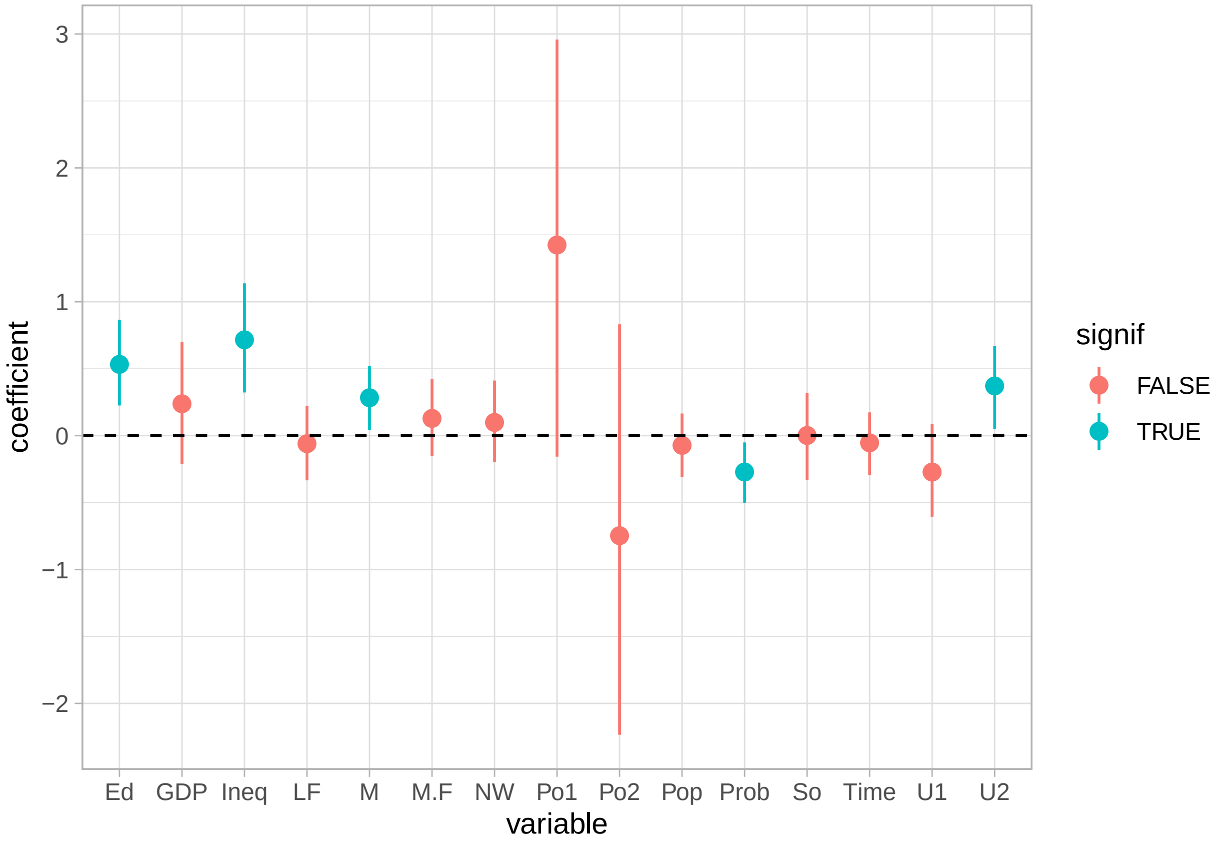
```
data("UScrime",package="MASS")
namvars <- names(UScrime)
```

## a) Fit a regression model

$y = X\beta + \epsilon$ using the g-prior with $g = n, \nu_0 = 2$ and $\sigma_0^2 = 1$. Obtain marginal posterior means and 95% confidence intervals for $\beta$, and compare to the least squares estimates. Describe the relationships between crime and the explanatory variables. Which variables seem strongly predictive of crime rates?

```
y = crime$y
X =as.matrix(crime[,-1]) # X = crime %>% select(-y) %>% as.matrix
n = dim(X)[1]
p = dim(X)[2]
g = n
nu0 = 2
s20 = 1
S = 1000
Hg = (g / (g + 1)) * X %*% solve(t(X) %*% X) %*% t(X)
SSRg = t(y) %*% (diag(1, nrow = n) - Hg) %*% y
s2 = 1 / rgamma(S, (nu0 + n) / 2, (nu0 * s20 + SSRg) / 2)
Vb = g * solve(t(X) %*% X) / (g + 1)
Eb = Vb %*% t(X) %*% y
E = matrix(rnorm(S * p, 0, sqrt(s2)), S, p)
beta = t(t(E %*% chol(Vb)) + c(Eb))
```

```
library(tidyr)
signif = apply(beta, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.5, 0.975)) %>%
  apply(MARGIN = 2, FUN = function(y) !(y[1] < 0 && 0 < y[3]))
beta_df = as.data.frame(beta) %>%
  gather(key = 'variable', val = 'coefficient') %>%
  mutate(signif = signif[variable])
ggplot(beta_df, aes(x = variable, y = coefficient, color = signif)) +
  stat_summary(fun.y=mean,fun.ymin=function(y)quantile(y,probs=c(0.025)),fun.ymax=function(y)quantile(y
  geom_hline(yintercept = 0, lty = 2)+theme_light()
```



Looks like Ed (mean years of schooling), Ineq (Income inequality), M (percentage of males aged 14-24), Prob (probability of imprisonment), and U2 (unemployment rate of urban males 35-39).

## b) Lets see how well regression models can predict crime rates based on the

X-variables. Randomly divide the crime roughly in half, into a training set $\{y_{tr}, X_{tr}\}$ and a test set $\{y_{te}, X_{te}\}$

```
y = crime$y
X =as.matrix(crime[,-1]) # X = crime %>% select(-y) %>% as.matrix
set.seed(1) # Reproducible!
train_i = sample.int(length(y), size = round(length(y) / 2), replace = FALSE)
ytr = y[train_i]
Xtr = X[train_i, ]
yte = y[-train_i]
Xte = X[-train_i, ]
```

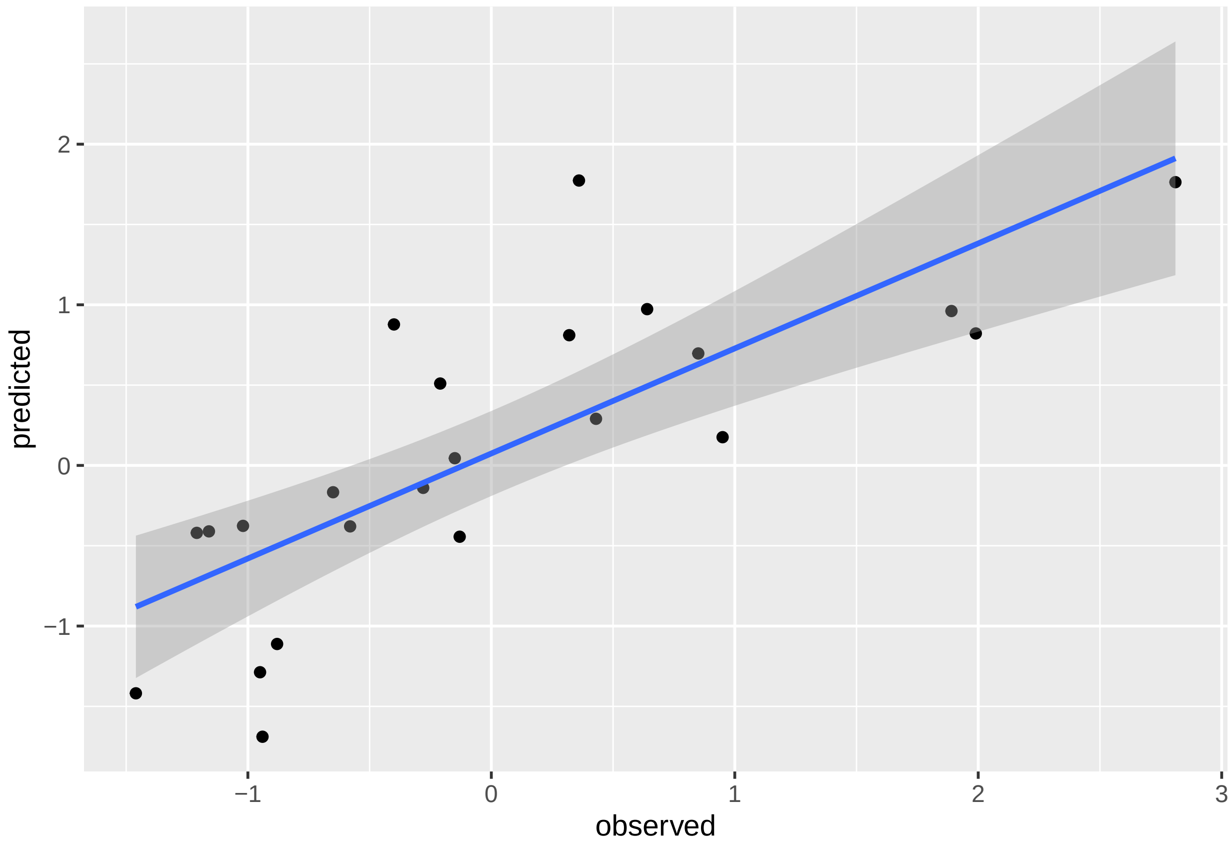
### i. Using only the training set, obtain least squares regression coefficients $\hat{\beta}_{ols}$.

Obtain predicted values for the test data by computing $\hat{y}_{ols} = \mathbf{X}_{te}\hat{\beta}_{ols}$. Plot $\hat{y}_{ols}$ versus $y_{te}$ and compute the prediction error $\frac{1}{n_{te}} \sum (y_{i,te} - \hat{y}_{i,ols})^2$.

```
beta_ols = solve(t(Xtr) %*% Xtr) %*% t(Xtr) %*% ytr # From 9.1
beta_ols
```

```
##             [,1]
## M      0.19419790
## So     0.20624856
## Ed     0.64334998
## Po1    0.30590821
## Po2    0.44725441
## LF    -0.09198965
## M.F    0.03537926
## Pop    0.08440462
## NW    -0.01227990
## U1    -0.03278782
## U2     0.15494093
## GDP    0.10455812
## Ineq   0.76691166
## Prob  -0.25823362
## Time   0.05938070
```

```
y_ols = Xte %*% beta_ols
ols_df = data.frame(observed = yte,predicted = y_ols)
ggplot(ols_df, aes(x = observed, y = predicted)) +geom_point() +
  geom_smooth(method = 'lm')
```



```
pred_error = sum((yte - y_ols)^2) / length(yte)
pred_error
```

```
## [1] 0.4880959
```

**ii. Now obtain the posterior mean $\beta_{Bayes} = E[\beta|y_{tr}]$ using the g-prior described above and the training data only.**

Obtain predictions for the test set $\hat{y}_{Bayes} = \mathbf{X}_{test}\hat{\beta}_{Bayes}$. Plot versus the test data, compute the prediction error, and compare to the OLS prediction error. Explain the results.

```r
y = ytr
X = Xtr
n = dim(X)[1]
p = dim(X)[2]
g = n
nu0 = 2
s20 = 1
S = 1000
Hg = (g / (g + 1)) * X %*% solve(t(X) %*% X) %*% t(X)
SSRg = t(y) %*% (diag(1, nrow = n) - Hg) %*% y
s2 = 1 / rgamma(S, (nu0 + n) / 2, (nu0 * s20 + SSRg) / 2)
Vb = g * solve(t(X) %*% X) / (g + 1)
Eb = Vb %*% t(X) %*% y
E = matrix(rnorm(S * p, 0, sqrt(s2)), S, p)
beta = t(t(E %*% chol(Vb)) + c(Eb))
beta_bayes = as.matrix(colMeans(beta))
y_bayes = Xte %*% beta_bayes
bayes_df = data.frame(
  observed = yte,
  predicted = y_bayes
)
ggplot(ols_df, aes(x = observed, y = predicted)) +
  geom_point() +geom_smooth(method = 'lm')+theme_light()
pred_error = sum((yte - y_bayes)^2) / length(yte)
pred_error
```

At least when the seed is 1, there doesn't appear to be a major difference between the prediction errors.

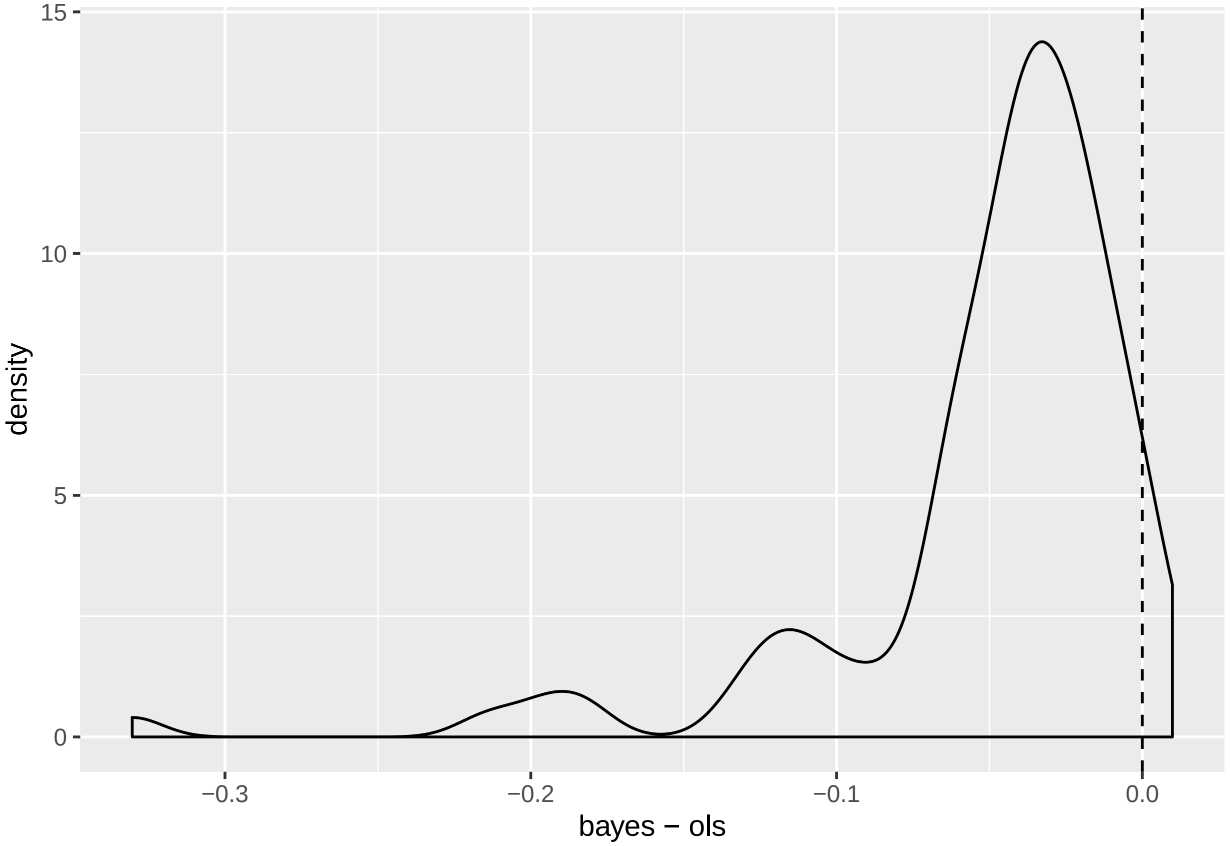
## c) Repeat the procedures in b) many times with different randomly generated

test and training sets. Compute the average prediction error for both the OLS and Bayesian methods.

```r
N = 100
set.seed(1)
pred_errors = t(sapply(1:N, function(i) {
  y = crime$y
  X =as.matrix(crime[,-1])
  train_i = sample.int(length(y), size = round(length(y) / 2), replace = FALSE)
  ytr = y[train_i]
  Xtr = X[train_i, ]
  yte = y[-train_i]
  Xte = X[-train_i, ]
  # OLS
  beta_ols = inv(t(Xtr) %*% Xtr) %*% t(Xtr) %*% ytr
  beta_ols
  y_ols = Xte %*% beta_ols
  pred_error_ols = sum((yte - y_ols)^2) / length(yte)
  # Bayes
  y = ytr
  X = Xtr
  n = dim(X)[1]
  p = dim(X)[2]
  g = n
  nu0 = 2
  s20 = 1
  S = 1000
  Hg = (g / (g + 1)) * X %*% inv(t(X) %*% X) %*% t(X)
  SSRg = t(y) %*% (diag(1, nrow = n) - Hg) %*% y
  s2 = 1 / rgamma(S, (nu0 + n) / 2, (nu0 * s20 + SSRg) / 2)
  Vb = g * inv(t(X) %*% X) / (g + 1)
  Eb = Vb %*% t(X) %*% y
  E = matrix(rnorm(S * p, 0, sqrt(s2)), S, p)
  beta = t(t(E %*% chol(Vb)) + c(Eb))
  beta_bayes = as.matrix(colMeans(beta))
  y_bayes = Xte %*% beta_bayes
  pred_error_bayes = sum((yte - y_bayes)^2) / length(yte)
  c(pred_error_ols, pred_error_bayes)
})) %>% as.data.frame
colnames(pred_errors) = c('ols', 'bayes')
```

Here's a plot of the density of $\text{err}_{\text{Bayes}} - \text{err}_{\text{ols}}$. If this is less than 0, then the Bayes estimator did better than the OLS estimator:

```
pred_diff = pred_errors %>% transmute(`bayes - ols` = bayes - ols)
ggplot(pred_diff, aes(x = `bayes - ols`)) +
  geom_density() + geom_vline(xintercept = 0, lty = 2)
```



```
mean(pred_errors$bayes < pred_errors$ols)
```

```
## [1] 0.96
```

For 100 samples, 96% of the time, the predictive error using the Bayes estimators is less than the predictive error using the OLS estimators.