Lab 6

PSTAT 115, Fall 2023

November, 2023

This lab will focus on the following topics:

- Single-parameter normal-normal model
- Multi-parameter normal-normal model
- Grid approximation to the posterior distribution
- Sampling from the joint posterior distribution

Review of single-parameter normal-normal model

Unknown μ, known σ^2

Prior:

$$\mu \sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right)$$

Likelihood:

$$Y_i \sim N\left(\mu, \sigma^2\right)$$

Posterior:

$$\mu | y_1, \ldots y_n \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

$$\text{where } \kappa_n = \kappa_0 + n \text{ and } \mu_n = \frac{\left(\kappa_0/\sigma^2\right)\mu_0 + \left(n/\sigma^2\right)\overline{y}}{\kappa_0/\sigma^2 + n/\sigma^2} = \frac{\kappa_0\mu_0 + n\overline{y}}{\kappa_n}$$

Discussion:

What is the meaning of κ_0 (consider pseudo-counts)?

How does the posterior parameters κ_n and μ_n relate to the prior parameters κ_0 and μ_0?

How would you interpret κ_n and μ_n?

Known μ, unknown σ^2

Prior:

$$p\left(\sigma^2\right) \propto \frac{1}{\sigma^2}$$

Likelihood:

$$Y_i \sim N\left(\mu, \sigma^2\right)$$

Posterior:

$$p\left(\sigma^2 | y\right) \propto \left(\sigma^2\right)^{-n/2-1} \exp\left\{-\sum_{i=1}^{n}(y_i-\mu)^2/\left(2\sigma^2\right)\right\}$$

Actually, when μ is known, the conjugate prior for σ^2 is an inverse Gamma distribution:

$$\sigma^2 | \alpha, \beta \sim \text{ Inv-gamma } (\alpha, \beta)$$

$$P(z|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} z^{-\alpha-1} \exp\left(-\frac{\beta}{z}\right).$$

For the posterior we get another inverse Gamma:

$$\sigma^2 | \mu, \alpha, \beta, y_1, \ldots y_n \propto \left(\sigma^2\right)^{-(\alpha+\frac{n}{2})-1} \exp\left(-\frac{\beta + \frac{1}{2}\sum(y_i-\mu)^2}{\sigma^2}\right)$$

$$\propto (\sigma^2)^{-\alpha_{post}-1} \exp\left(-\frac{\beta_{post}}{\sigma^2}\right)$$

The posterior is

$$\sigma^2 | \mu, \alpha, \beta, y_1, \ldots y_n \sim \text{Inv-gamma}(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum(y_i-\mu)^2)$$

Multi-parameter normal-normal model

In the normal model we typically factorize the prior distribution

$$p\left(\mu, \sigma^2\right) = p(\mu|\sigma^2) p\left(\sigma^2\right)$$

Specifically:

$$\sigma^2 \sim \text{ Inv-gamma } \left(\nu_0/2, \nu_0\sigma_0^2/2\right)$$

$$\mu | \sigma^2 \sim \text{ Normal } \left(\mu_0, \sigma^2/\kappa_0\right)$$

$$Y_1, \ldots, Y_n | \mu, \sigma^2 \sim \text{ i.i.d. Normal } \left(\mu, \sigma^2\right)$$

ν_0 is a prior sample size and σ_0^2 is the prior sample variance

κ_0 is a prior sample size and μ_0 is the prior sample mean

Posterior:

$$p(\mu | y_1, \ldots y_n, \sigma^2) \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

$$\kappa_n = \kappa_0 + n \text{ and } \mu_n = \frac{\left(\kappa_0/\sigma^2\right)\mu_0 + \left(n/\sigma^2\right)\overline{y}}{\kappa_0/\sigma^2 + n/\sigma^2} = \frac{\kappa_0\mu_0 + n\overline{y}}{\kappa_n}$$

$$\sigma^2 | y_1, \ldots y_n \sim \text{Inv-gamma}\left(\nu_n/2, \nu_n\sigma_n^2/2\right)$$

$$\nu_n = \nu_0 + n$$

$$\sigma_n^2 = \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_n}\left(\overline{y}-\mu_0\right)^2\right]$$

Grid approximation to the posterior distribution

First of all we set up the parameters based on the above formulas.

```r
## prior mean for mu and prior counts for mu
mu0 <- 1.9
k0 <- 1

## prior mean for variance and prior counts for variance
s20 <- 0.010
nu0 <- 1

## sufficient statistics are sample mean and sample variance
y <- c(1.64, 1.7, 1.72, 1.74, 1.82, 1.82, 1.82, 1.9, 2.08)
n <- length(y)
ybar <- mean(y)
s2 <- var(y)

## posterior parameters, see the formula above
kn <- k0 + n
nun <- nu0 + n
mun <- (k0 * mu0 + n * ybar) / kn
s2n <- (nu0*s20 + (n-1)*s2  + k0*n / kn * (ybar - mu0)^2) / nun
```

Now we need to learn several basic functions to facilitate our plots.

(1) The `expand.grid()` function creates a grid based on two vectors by justaposing a pair of values. The value from the first vector changes first, then the value from the second vector will change.

```r
## example for expand.grid
grid <- as_tibble(expand.grid(seq(1, 3, by = 1), seq(0, 2, by=1)))
colnames(grid) <- c('x', 'y')
grid
```

```
## # A tibble: 9 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     0
## 2     2     0
## 3     3     0
## 4     1     1
## 5     2     1
## 6     3     1
## 7     1     2
## 8     2     2
## 9     3     2
```

(2) The `Vectorize()` function is a wrapper for an arbitrary function. The resulting function can be applied to each row for a tibble object.

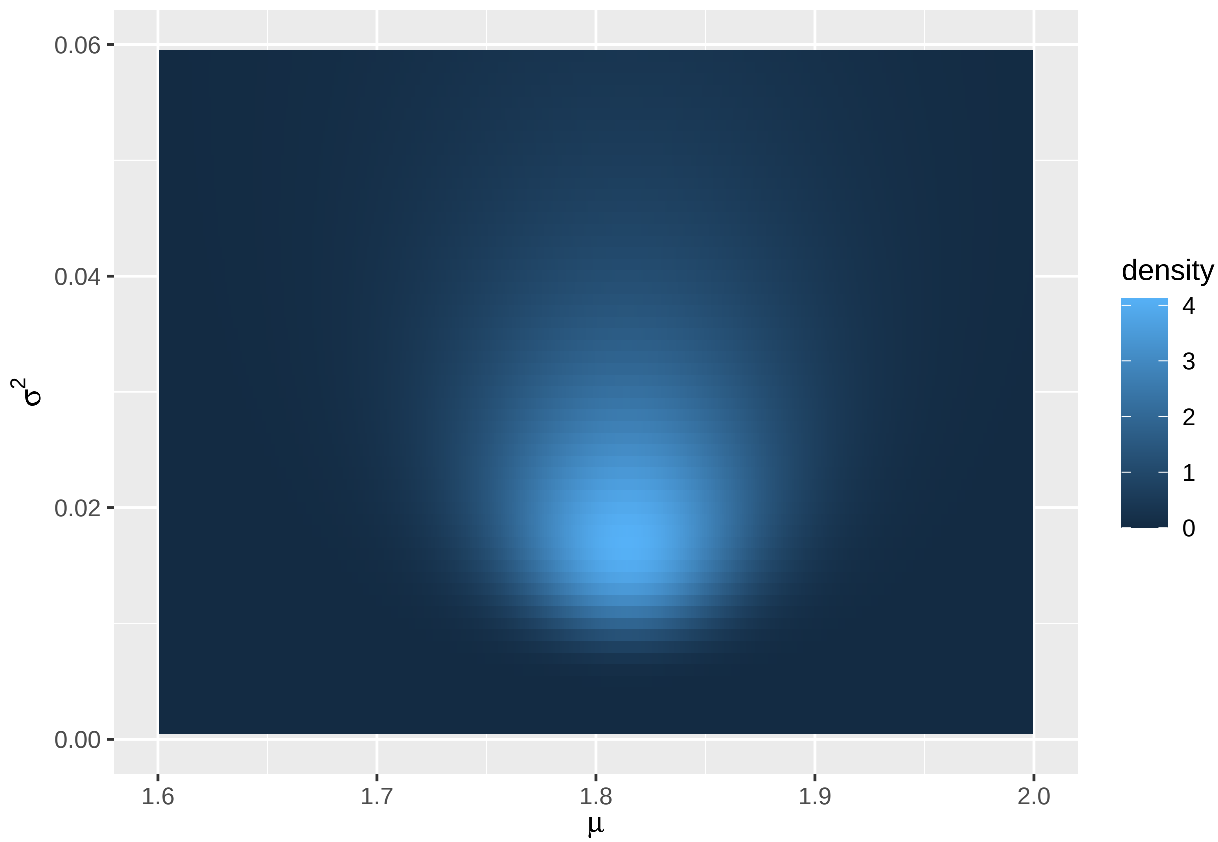
```
## example of Vectorize()
sum <- Vectorize(function (a, b) {
  a + b
})

grid %>% mutate("Sum" = sum(x, y))
```

```
## # A tibble: 9 x 3
##       x     y   Sum
##   <dbl> <dbl> <dbl>
## 1     1     0     1
## 2     2     0     2
## 3     3     0     3
## 4     1     1     2
## 5     2     1     3
## 6     3     1     4
## 7     1     2     3
## 8     2     2     4
## 9     3     2     5
```

Armed with the above new functions, let us go about plotting the posterior joint density.

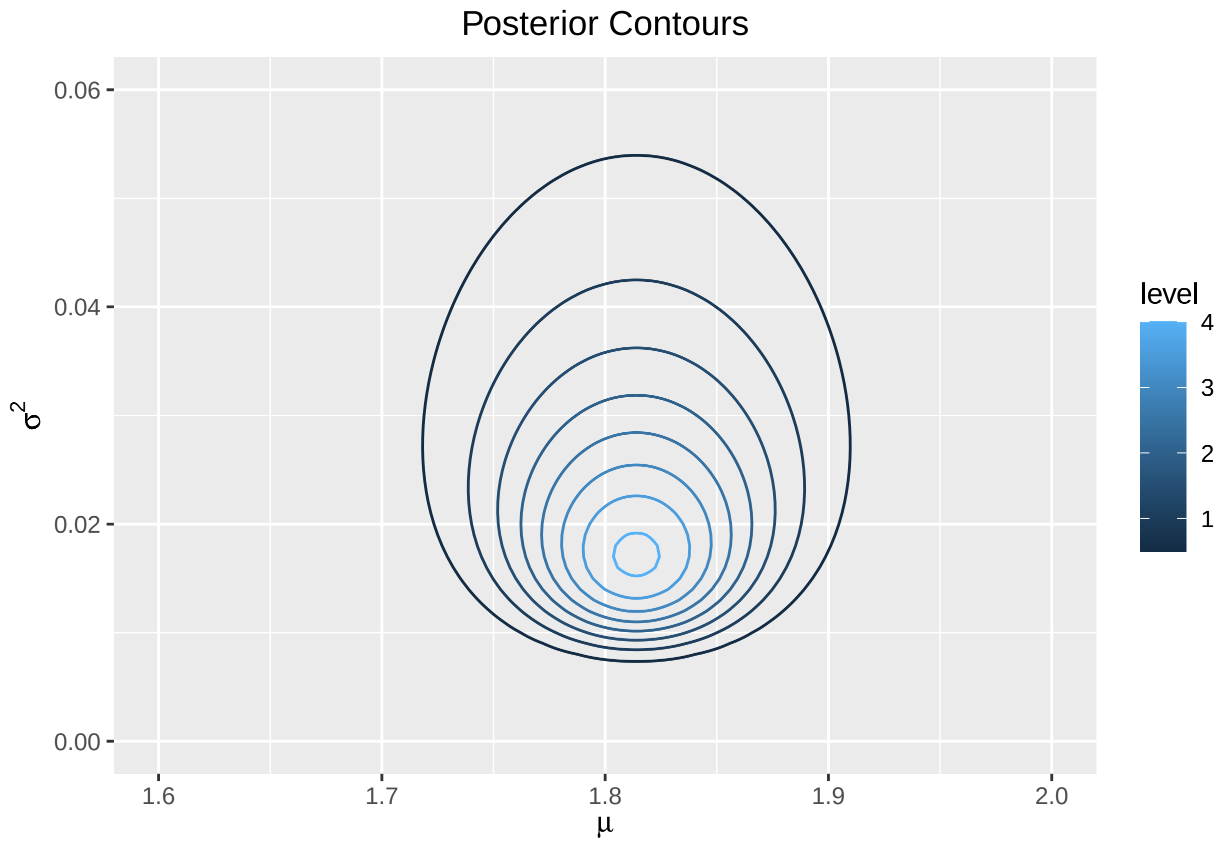
```
## create the grid on which the posterior joint distribution we are interested in
grid <- as_tibble(expand.grid(seq(1.6, 2.0, by=0.001), seq(0, 0.06, by=0.001)))
colnames(grid) <- c("mu", "s2")

## create the wrapped function to be applied to each row in the tibble
normal_posterior <- Vectorize(function(mu, sigma2) {

  ## likelihood times prior
  prod(dnorm(y, mu, sqrt(sigma2))) *
    dnorm(mu, mu0, sqrt(sigma2/k0)) *
    dgamma(1/sigma2, nu0/2, nu0/2*s20)

})

## applied the wrapped function to each row of grid and
## plot the density using the geom_raster function,
## which fill each location based on the corresponding density value

grid %>% mutate(density = normal_posterior(mu, s2)) %>% ggplot() +
  geom_raster(aes(mu, s2, fill=density)) +
  xlim(c(1.6, 2)) + ylim(c(0, 0.06)) +
  xlab(expression(mu)) + ylab(expression(sigma^2))
```

```
## Warning: Removed 920 rows containing missing values (‘geom_raster()‘).
```



Contour Plot

```
grid %>% mutate(density = normal_posterior(mu, s2)) %>%
  ggplot() + geom_contour(aes(mu, s2, z=density, colour=stat(level))) +
  xlim(c(1.6, 2)) + ylim(c(0, 0.06)) +
  xlab(expression(mu)) + ylab(expression(sigma^2)) +
  ggtitle("Posterior Contours") +
  theme(plot.title = element_text(hjust = 0.5))
```



Sampling from the joint posterior distribution

$$p\left(\mu, \sigma^2 | y_1, \dots y_n\right) = p(\mu | y_1, \dots y_n, \sigma^2) p\left(\sigma^2 | y_1, \dots y_n\right)$$

The first one is the single-parameter normal model with σ^2 known.

$$p(\mu | y_1, \dots y_n, \sigma^2) \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

For the second one we need to marginalize σ^2 by integrating out μ.

$$\begin{aligned} p\left(\sigma^2 | y_1, \dots, y_n\right) &\propto p\left(\sigma^2\right) p\left(y_1, \dots, y_n | \sigma^2\right) \\ &= p\left(\sigma^2\right) \int p\left(y_1, \dots, y_n | \mu, \sigma^2\right) p(\mu | \sigma^2) d\theta \end{aligned}$$

Result:

$$\begin{aligned} 1/\sigma^2 | y_1, \dots, y_n &\sim \text{Gamma}\left(\nu_n/2, \nu_n \sigma_n^2/2\right), \text{ where} \\ \nu_n &= \nu_0 + n \\ \sigma_n^2 &= \frac{1}{\nu_n}\left[\nu_0 \sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_n}\left(\overline{y} - \mu_0\right)^2\right] \end{aligned}$$

Given $p(\mu|y_1, \ldots y_n, \sigma^2)$ and $p\left(\sigma^2|y_1, \ldots y_n\right)$ we have a simple 2-step algorithm for sampling from the joint posterior:

Step 1: Sample

$$\sigma^2 \sim \text{Inv-gamma}\left(\nu_n/2, \nu_n\sigma_n^2/2\right)$$

Step 2: Sample

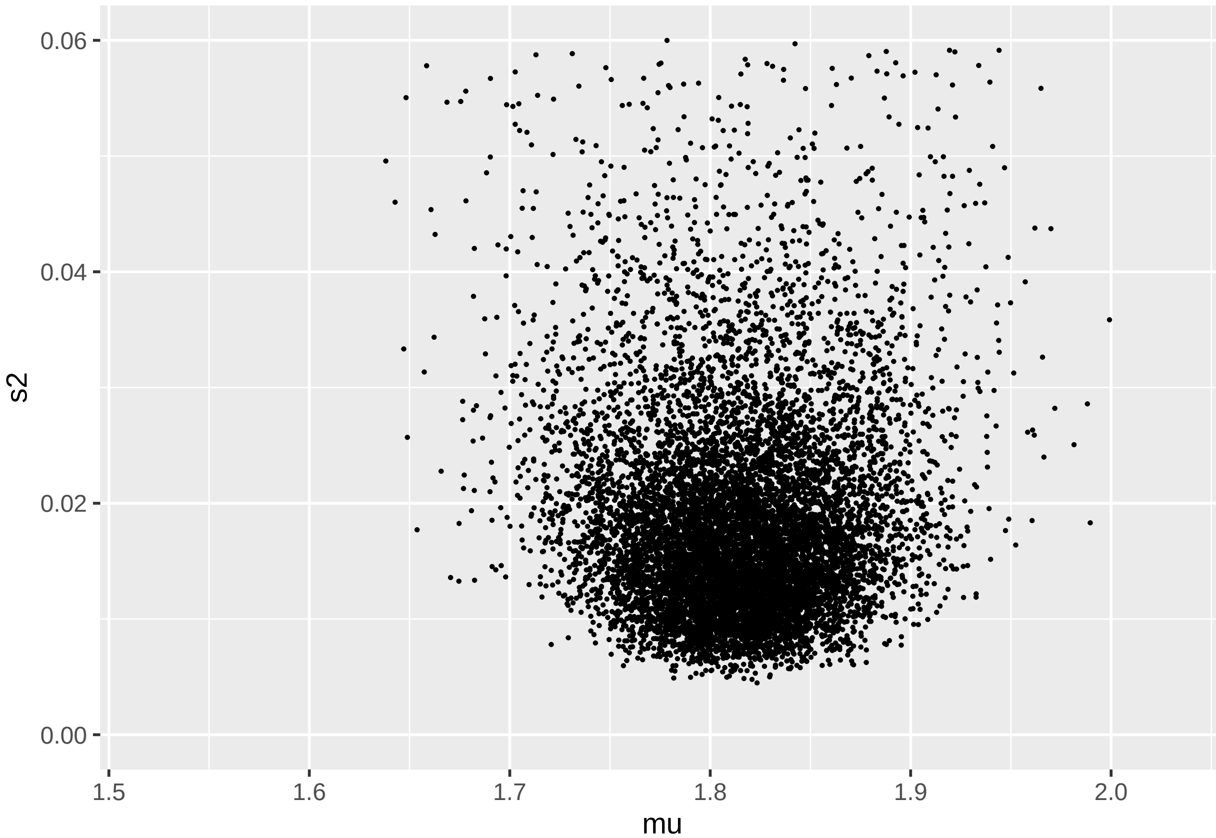
$$\mu \sim N\left(\mu_n, \frac{\sigma^2}{\kappa_n}\right)$$

We again set the parameters first.

```
## prior mean for mu and prior counts for mu
mu0 <- 1.9
k0 <- 1

## prior mean for variance and prior counts for variance
s20 <- 0.010
nu0 <- 1

## sufficient statistics are sample mean and sample variance
y <- c(1.64, 1.7, 1.72, 1.74, 1.82, 1.82, 1.82, 1.9, 2.08)
n <- length(y)
ybar <- mean(y)
s2 <- var(y)

## posterior parameters
kn <- k0 + n
nun <- nu0 + n
mun <- (k0 * mu0 + n * ybar) / kn
s2n <- (nu0*s20 + (n-1)*s2  + k0*n / kn * (ybar - mu0)^2) / nun
```

Now we get 10000 samples from the posterior joint distribution.

```
nsamps <- 10000

## step 1. s2 | y
s2_samps <- 1/rgamma(nsamps, nun/2, nun*s2n/2)

## step 2. mu | s2, y, remember rnorm takes sd not var!
mu_samps <- rnorm(nsamps, mun, sd = sqrt(s2_samps/kn))

## plot the samples
tibble(mu=mu_samps, s2=s2_samps) %>% ggplot() +
  geom_point(aes(x=mu, y=s2), size=.25) + ylim(c(0, 0.06))
```



After we have samples, we can evaluate some probabilities based on the Monte Carlo spirit. For example, what is the posterior probability that μ is greater than $\bar{y}$ and σ^2 is less than the sample variance?

```
mean(mu_samps > ybar & s2_samps < s2)
```

```
## [1] 0.3219
```

Now you have mastered the most difficult task so far and you are well-prepared for the fascinating MCMC techniques!