Applied Bayesian Modeling - module 3

Leontine Alkema

August 29, 2022

R code and output to reproduce results in the slides

1 Radon

Read in the data and process (little more than we need for this analysis, other variables will be used later on in course)

```
# house level data
d <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/srrs2.dat"), header=T, sep=

# deal with zeros, select what we want, make a fips (county) variable to match on
d <- d %>%
  mutate(activity = ifelse(activity==0, 0.1, activity)) %>%
  mutate(fips = stfips * 1000 + cntyfips) %>%
  dplyr::select(fips, state, county, floor, activity)

# county level data
cty <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/cty.dat"), header = T, s
cty <-
  cty %>%
  mutate(fips = 1000 * stfips + ctfips) %>%
  dplyr::select(fips, Uppm) %>%
  rename(ura_county = (Uppm))

dmn <- d %>%
  filter(state=="MN") %>% # Minnesota data only
  dplyr::select(fips, county, floor, activity) %>%
  left_join(cty)
```

```
## Joining, by = "fips"
```

```
head(dmn)
```

```
##    fips              county floor activity ura_county
## 1 27001 AITKIN                  1      2.2   0.502054
## 2 27001 AITKIN                  0      2.2   0.502054
## 3 27001 AITKIN                  0      2.9   0.502054
## 4 27001 AITKIN                  0      1.0   0.502054
## 5 27003 ANOKA                   0      3.1   0.428565
## 6 27003 ANOKA                   0      2.5   0.428565
```

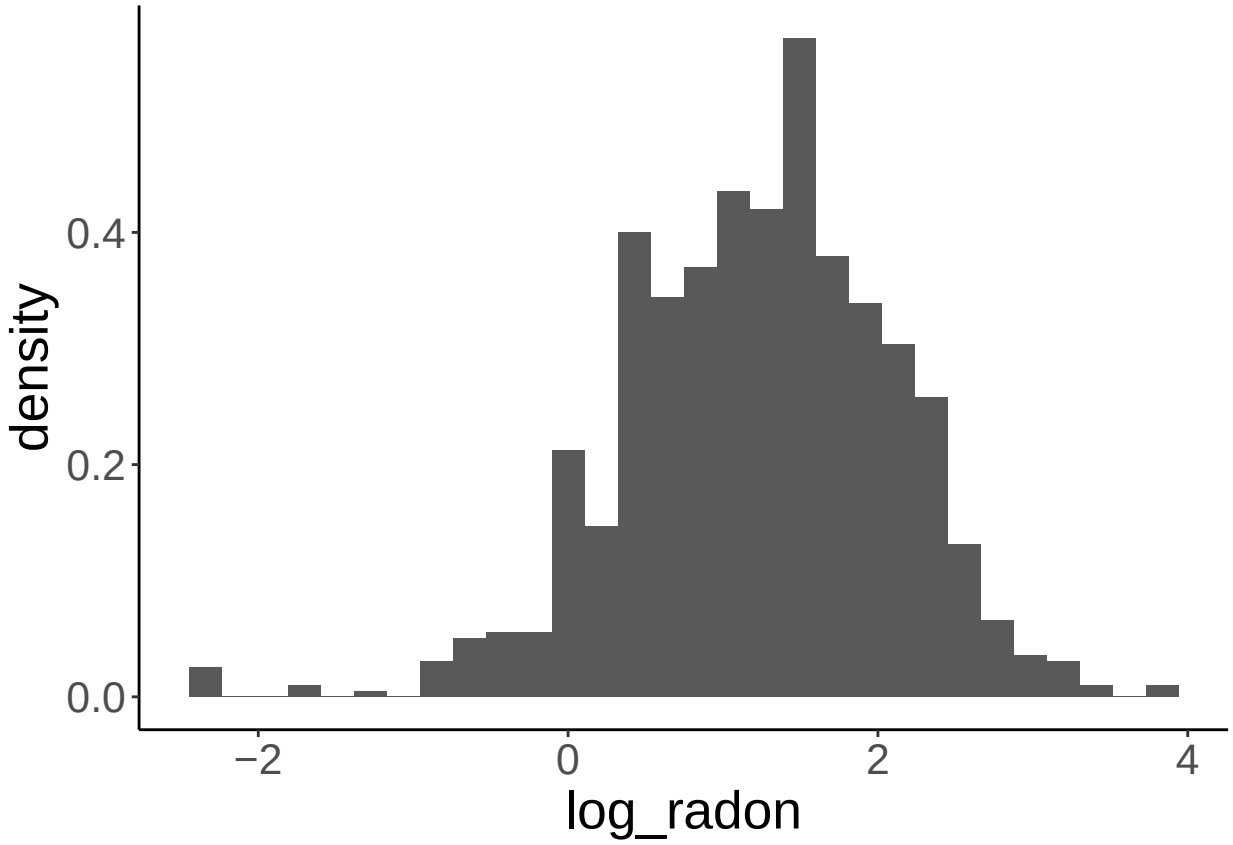
Data for estimating population mean

```
y <- log(dmn$activity)
```

Histogram of the data

```
tibble(log_radon = y) %>%
  ggplot(aes(x = log_radon)) +
  geom_histogram(aes(y=..density..)) +
  theme_classic() +
  theme(text = element_text(size = 20))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2 Inference

Goal: Estimate μ, assume a value for σ

Information from the data

```
# data
ybar <- mean(y)
sd.y <- sd(y)
n <- length(y)
```

Fix sigma

```
sigma <- sd.y
# sd for ybar follows from sigma
sd.ybar <- sigma/sqrt(n)
```

Fix prior mean and prior sd

```
mu0 <- 0 # prior mean
sigma.mu0 <- 1  # prior sd

# other option used in slides
#mu0 <- -ybar # prior mean
#sigma.mu0 <- sd.ybar
```

Then we can obtain posterior mean and variance

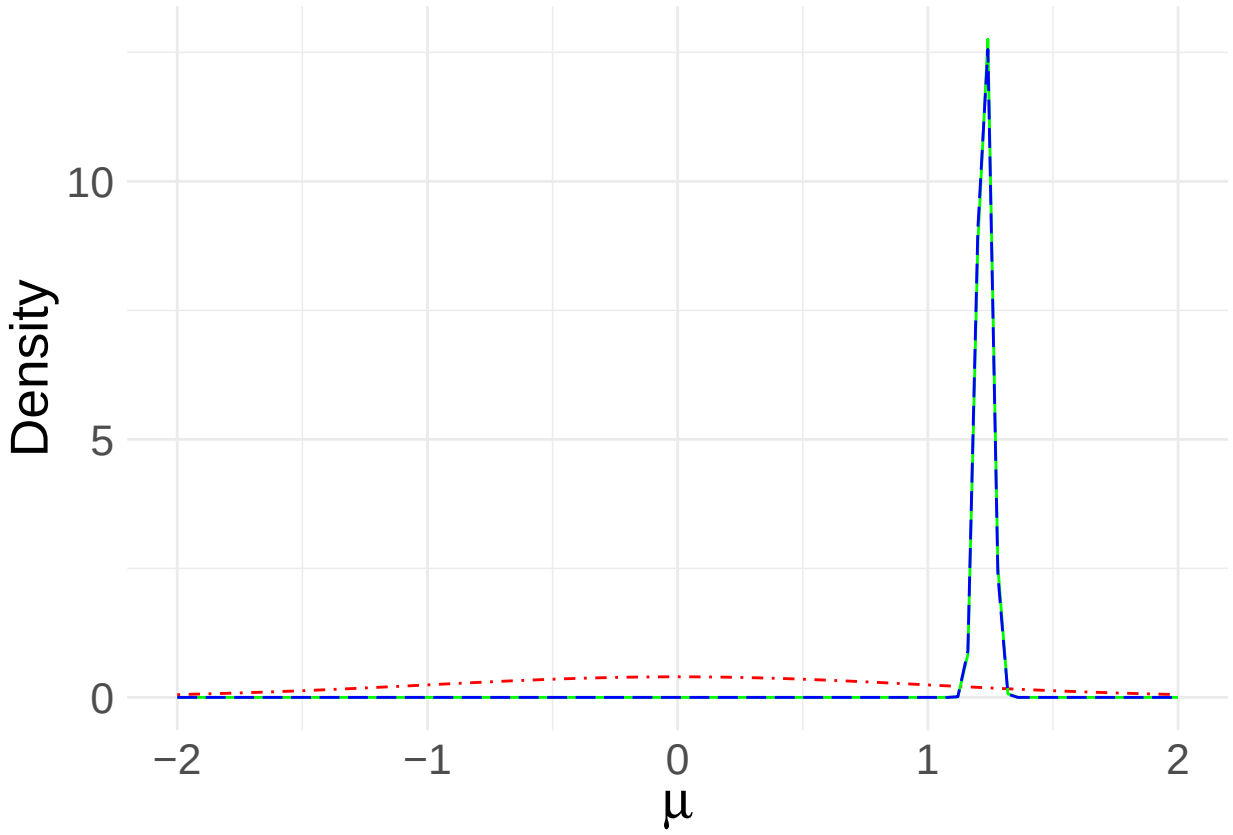
```
mupost.mean <- (mu0/(sigma.mu0^2) + n*ybar/(sigma^2))/(1/(sigma.mu0^2) + n/(sigma^2))
mupost.sd <- sqrt(1/(1/(sigma.mu0^2)+n/(sigma^2)))
```

2.1 Plot prior, likelihood, and posterior

Different ways to go about plotting, here's one using functions:

```
prior_dens <- function(x) dnorm(x, mean = mu0 , sd = sigma.mu0)
post_dens <- function(x) dnorm(x, mean = mupost.mean, sd = mupost.sd )
like <- function(x) dnorm(x,  mean = ybar, sd = sd.ybar)

ggplot(NULL, aes(c(-2,2))) +
  geom_line(stat = "function", fun = prior_dens, color = "red", linetype = "dotdash") +
  geom_line(stat = "function", fun = like, linetype = "solid", color = "green") +
  geom_line(stat = "function", fun = post_dens, linetype = "longdash", color = "blue") +
  theme_minimal() +
  ylab("Density") +
  xlab(expression(mu)) +
  theme(
    legend.position = "top",
    legend.title = element_blank(),
    text = element_text(size = 20)
  )
```



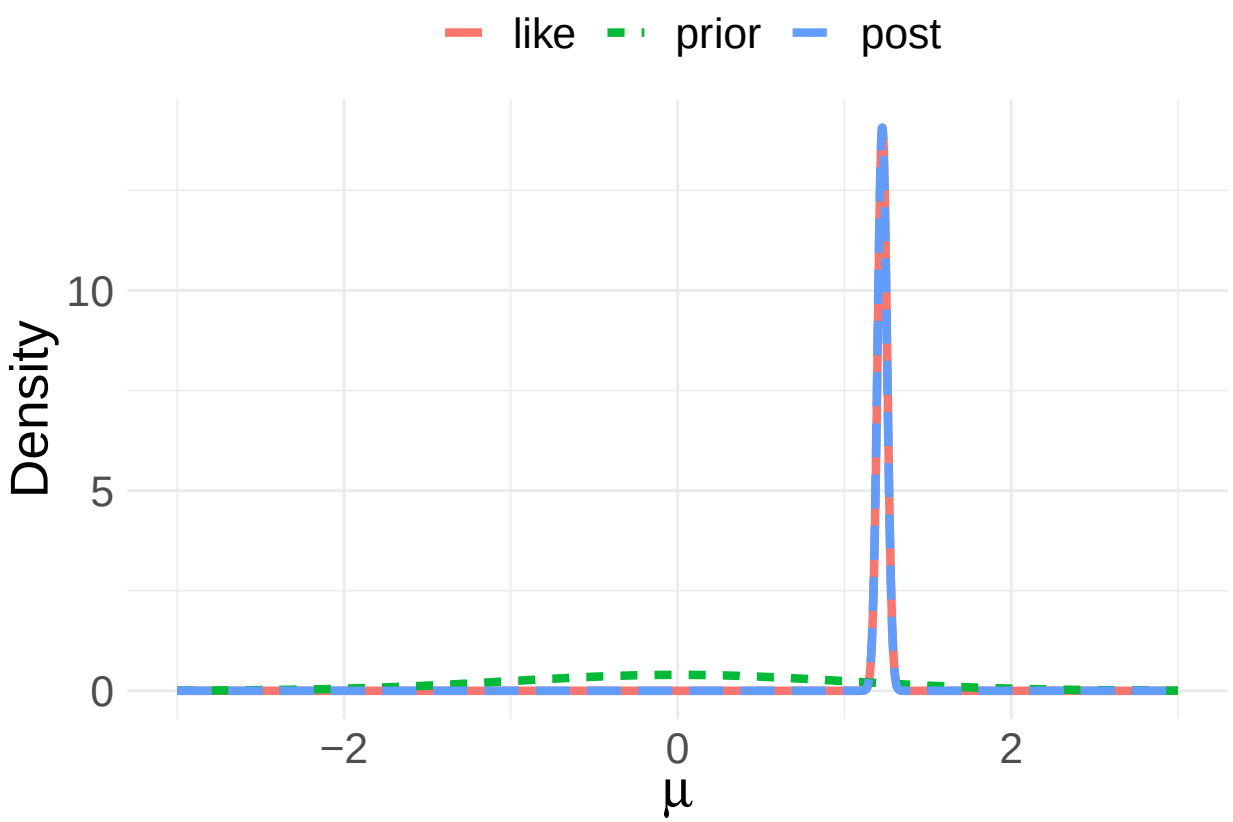

```
#  geom_area(stat = "function", fun = prior_dens) + #, fill = "red")
```

In the slides, I used ones where I calculate the densities for a grid, save that in a tibble toplot, and plot the tibble

```
# hard-coded grid
# mugrid <- seq(1, 1.5, length.out = 3000)
# or, based on parameters
mugrid <- seq(
   min(mu0 - 3*sigma.mu0, mupost.mean - 3*mupost.sd, ybar - 3*sd.ybar),
   max(mu0 + 3*sigma.mu0, mupost.mean + 3*mupost.sd, ybar + 3*sd.ybar),
  length.out = 3000)
prior.dens <- dnorm(x = mugrid, mean = mu0 , sd = sigma.mu0)
like.dens <- dnorm(x = mugrid, mean = ybar, sd = sd.ybar)
post.dens <- dnorm(x = mugrid, mean = mupost.mean, sd = mupost.sd)
toplot <- tibble(
  dens = c(prior.dens,  like.dens, post.dens),
  dtype = rep(c("prior", "like", "post"), each = length(mugrid)),
  mugrid = rep(mugrid, 3))

toplot %>%
  mutate(dtype = factor(dtype, levels = c("like", "prior", "post"))) %>%
  ggplot(aes(
    x = mugrid,
    y = dens,
    col = dtype,
    lty = dtype
```

```
))   +
geom_line(size = 1.5) +
theme_minimal() +
ylab("Density") +
xlab(expression(mu)) +
theme(
  legend.position = "top",
  legend.title = element_blank(),
  text = element_text(size = 20)
)
```



2.2 Summarize the posterior

Bayesian inference

```
mupost.mean # posterior mean
```

```
## [1] 1.226465
```

```
qnorm(0.5, mean = mupost.mean, sd = mupost.sd) # posterior median
```

```
## [1] 1.226465
```

```r
qnorm(c(0.025, 0.975), mean = mupost.mean, sd = mupost.sd) # 95% quantile-based CI
```

```
## [1] 1.170903 1.282027
```

Frequentist inference for pop mean with know variance

```r
ybar
```

```
## [1] 1.227451
```

```r
ybar + qnorm(c(0.025, 0.975))*sd.ybar
```

```
## [1] 1.171867 1.283036
```

3 Extra

Plot normal pdf and cdf

```r
par(mfrow = c(1,2))
curve(pnorm(x), xlim = c(-3,3), ylab = "F(x)")
curve(dnorm(x), xlim = c(-3,3), ylab = "p(x)")
```