Applied Bayesian modeling - HW1

Álvaro J. Castro Rivadeneira

September 12, 2022

Score: The maximum number of points in this HW is 15 points, with 3 points extra credit. For calculating a final HW grade, the points will be rescaled to a maximum score of (15+3)/15*100% = 120%.

What to hand in: For exercise 2, we need an Rmd and a knitted pdf. You may hand in the answer to exercise 1 in a different output form as long as it's legible (i.e., no difficult-to-read picture of handwritten notes).

Exercise 1: Breast cancer and mammogram screening [5 pts]

This exercise is about the material in module 2.

Background information: Gerd Gegerenzer explained to 24 physicians:

- For early detection of breast cancer, women are encouraged to have routine screening, even if they have no symptoms.
- Imagine you conduct such screening using mammography
- The following information is available about asymptomatic women aged 40 to 50 in your region who have mammography screening:
  - The probability an asymptomatic woman has breast cancer is 0.8%.
  - If she has breast cancer, the probability is 90% that she has a positive mammogram.
  - If she does not have breast cancer, the probability is 7% that she still has a positive mammogram.
- Suppose a woman has a positive mammogram: What is the probability she actually has breast cancer?
- Physicians' answers ranged from about 1% to about 90%.

Use Bayes' rule to obtain the probability that a woman with a postive mammogram has breast cancer, using the information provided above. Show working, meaning to write out how you obtained probabilities that are not given in the information.

Answer

A is the probability that an asymptomatic woman has breast cancer A' is the probability that an asymptomatic woman does not have breast cancer B is the probability that a woman gets a positive result in her mammogram The probability that a woman has breast cancer given that she has a positive mammogram is $P(A|B)$, and the formula for solving it is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The probability that a woman has a positive mammogram given that she has breast cancer was already given and is:

$$P(B|A) = 0.9$$

The probability that an asymptomatic woman has breast cancer was already given and is:

$$P(A) = 0.008$$

The probability that a woman gets a positive result in her mammogram is given by:

$$P(B) = P(A) \cdot P(B|A) + P(A') \cdot P(B|A')$$

The probability that a woman does not have breast is given by:

$$P(A') = 1 - P(A) = 1 - 0.008 = 0.992$$

The probability that a woman gets a positive result in her mammogram if she does not have breast cancer was already given and is:

$$P(B|A') = 0.07$$

Thus, the probability that a woman gets a positive result in her mammogram is:

$$P(B) = 0.008 \cdot 0.9 + 0.992 \cdot 0.07 = 0.07664$$

```
pB = (0.008 * 0.9) + (.992 * .07)
pB
```

```
## [1] 0.07664
```

Finally, the probability that a woman has breast cancer given that she has a positive mammogram is:

$$P(A|B) = \frac{0.9 \cdot 0.008}{0.07664} = 0.09394572 \approx 9.4\%$$

```
pA.B = (0.9 * 0.008) / 0.07664
pA.B
```

```
## [1] 0.09394572
```

Thus, the probability that a woman with a postive mammogram has breast cancer is $\approx 9.4\%$.

Exercise 2: posteriors when everything's normal

This exercise is about module 3. You may find the R notebook with module 3 helpful.

We continue with the radon data set. (Note that I read in and process the data in the HW Rmd but don't print out the code).

We will carry out Bayesian inference assuming a normal likelihood and prior:

$y_i|\mu, \sigma^2 \sim N(\mu, \sigma^2)$, independent;
$\mu \sim N(m_0, s_{\mu 0}^2)$

with prior mean parameters $\mu_0 = 0$ and $s_{\mu 0} = 0.1$ and $\sigma = s\{y\}$.
$\bar{y}$ is given by the log-radon data.

Exercise 2a [5 pts, with additional 1 pt extra credit]

Use the radon data and obtain the posterior distribution $p(\mu|\mathbf{y})$, using the prior and likelihood as specified above. Use the posterior to produce the following outputs:

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

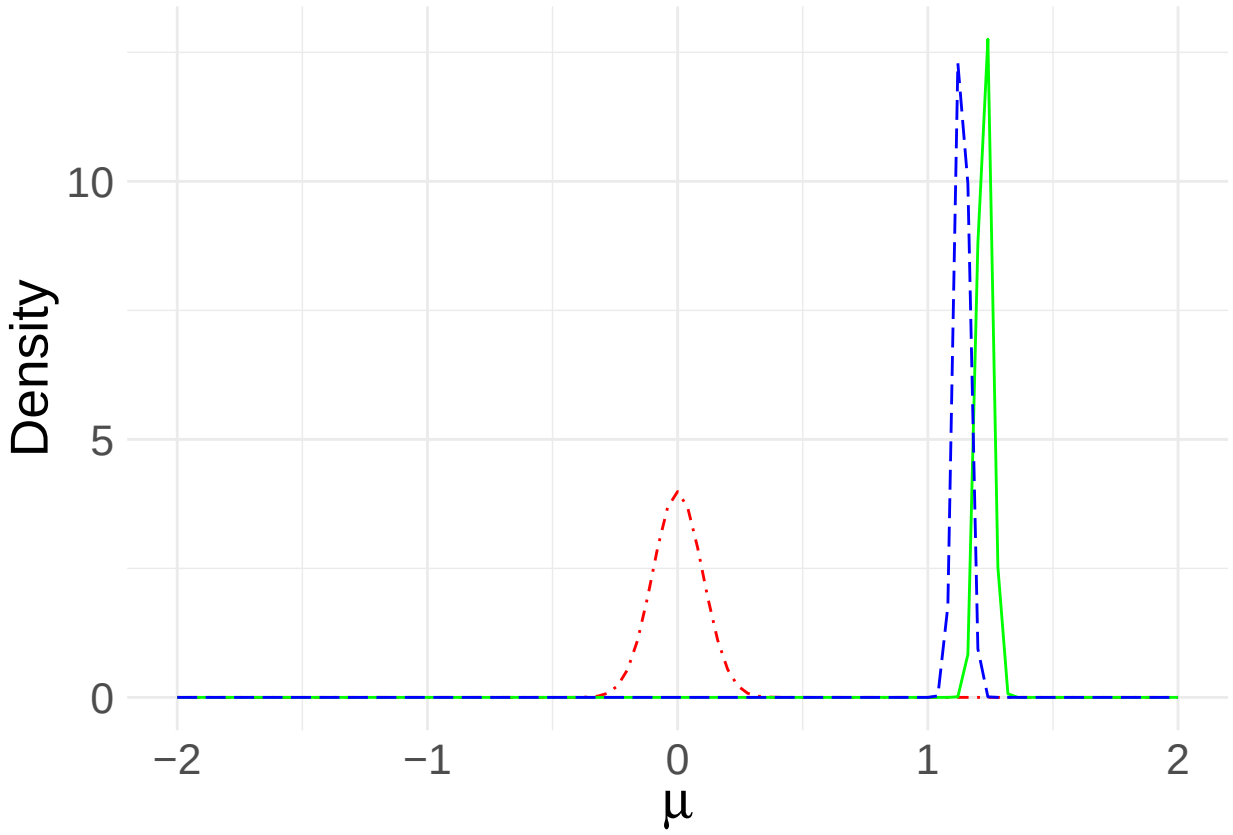
```
mu0 <- 0 # prior mean
sigma.mu0 <- 0.1  # prior sd
```

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)))
```

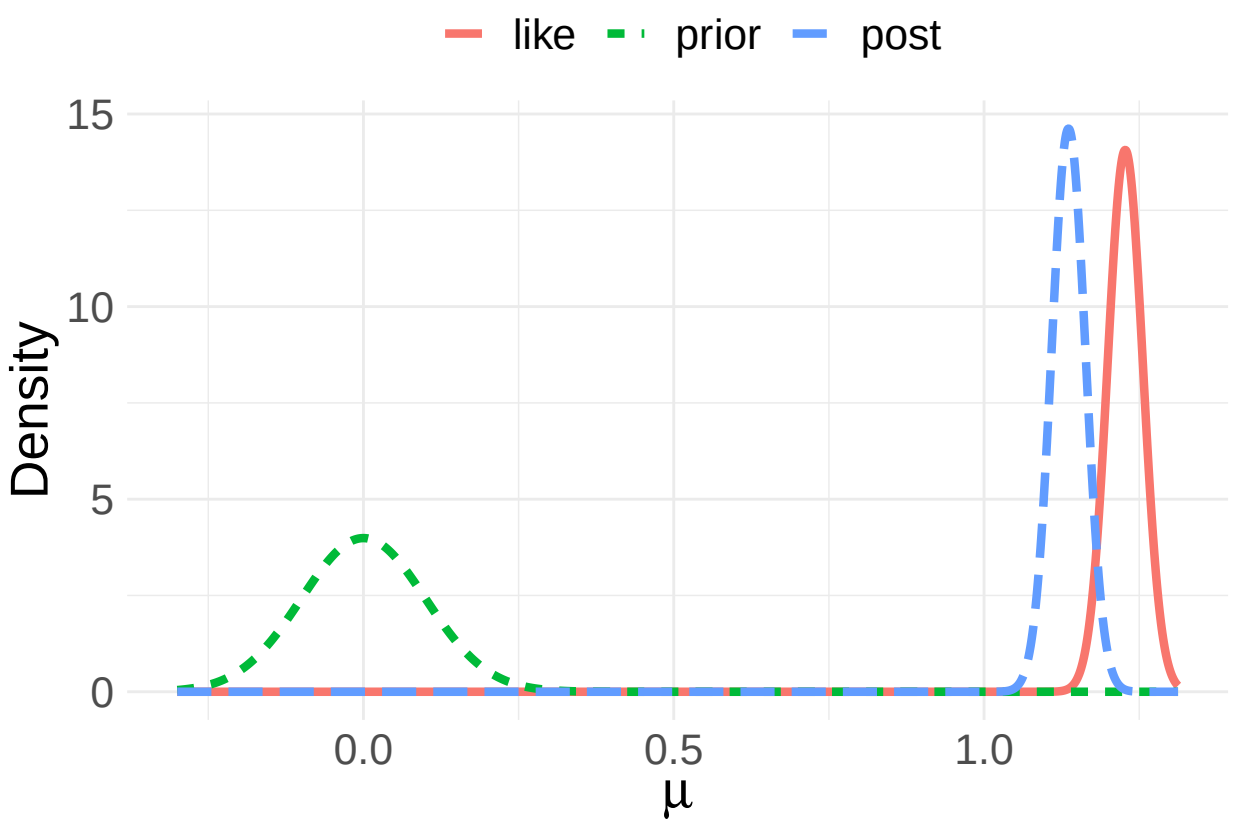
(i) one plot with the prior, posterior, and likelihood function;

```
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)
  )
```




```
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)
)
```



(ii) a point estimate, 95% credible interval, and 80% credible interval.

```
mupost.mean # posterior mean - point estimate
```

```
## [1] 1.136079
```

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

```
## [1] 1.136079
```

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

```
## [1] 1.082604 1.189554
```

```
qnorm(c(0.1, 0.9), mean = mupost.mean, sd = mupost.sd) # 80% quantile-based CI
```

```
## [1] 1.101113 1.171045
```

(iii) Interpretation of the 80% credible interval.

It is the posterior probability that μ is contained in the interval between 1.10 and 1.17 which corresponds to a quantile based division of its probability distribution between 0.1 and 0.9.

Extra credit question (1 pt): Can you calculate the probability that μ is greater than $\bar{y}$? If yes, report it. If not, why not?

$$P(\mu > \bar{y}) = 1 - P(\bar{y})$$

```
P.ybar = pnorm(ybar, mean = mupost.mean, sd = mupost.sd)
P.notybar = 1 - P.ybar
P.notybar
```

```
## [1] 0.0004055814
```

The probability that μ is greater than $\bar{y}$ is 0.0004056, which is almost zero. This makes sense since ybar is 1.227, which is greater than the 95% credible interval for the posterior distribution.

Exercise 2b [5pts]

Let's call the data set used so far data set 1. Suppose there is a second radon data set, referred to as data set 2, that has the same $\bar{y}$ and $s\{y\}$ as data set 1. What differs between the two data sets is that data set 2 has sample size $n = 4635$, which is 5 times the sample size of data set set 1 (with $n = 927$).

Obtain the posterior using data set 2, and produce the same outputs (i) and (ii) from exercise a. (No need to interpret the CI).

```
n <- 4635
```

Fix sigma

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

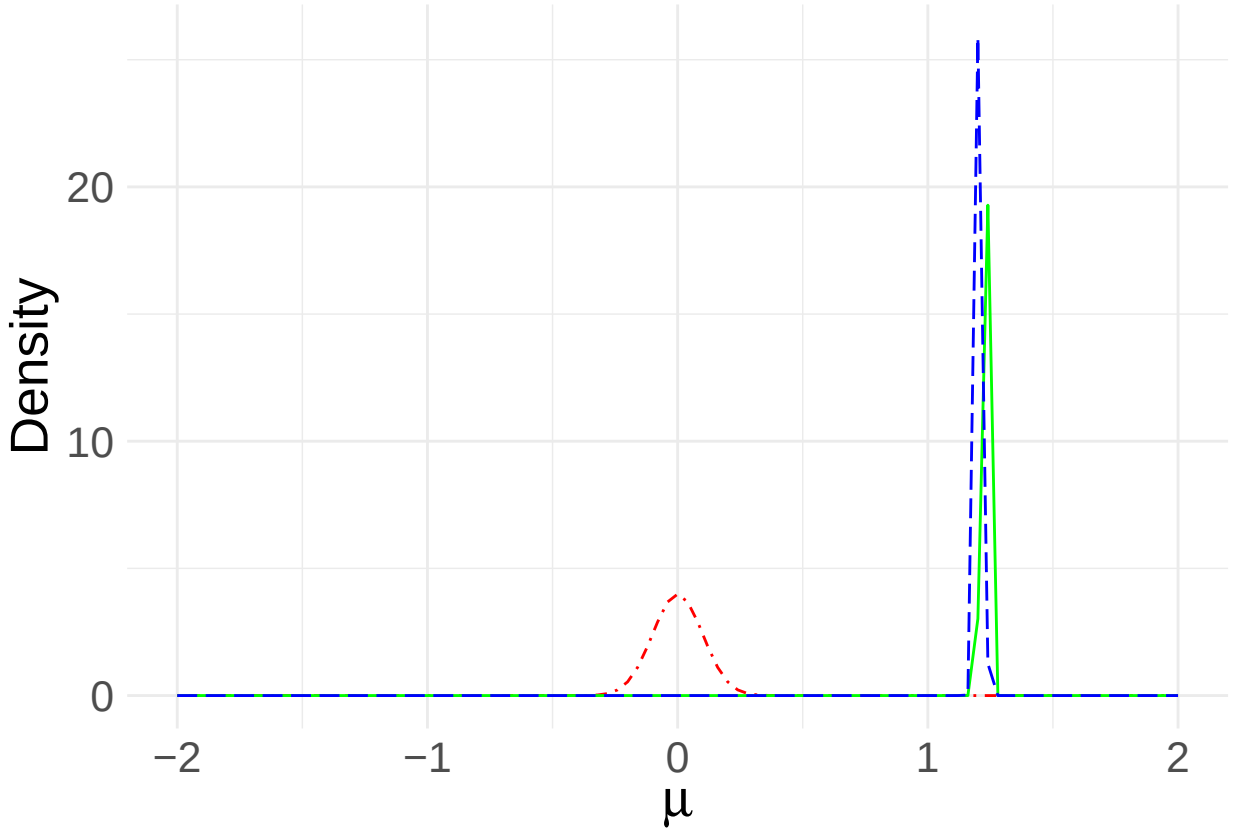
Then we can obtain posterior mean and variance

```
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)))
```

(i) one plot with the prior, posterior, and likelihood function;

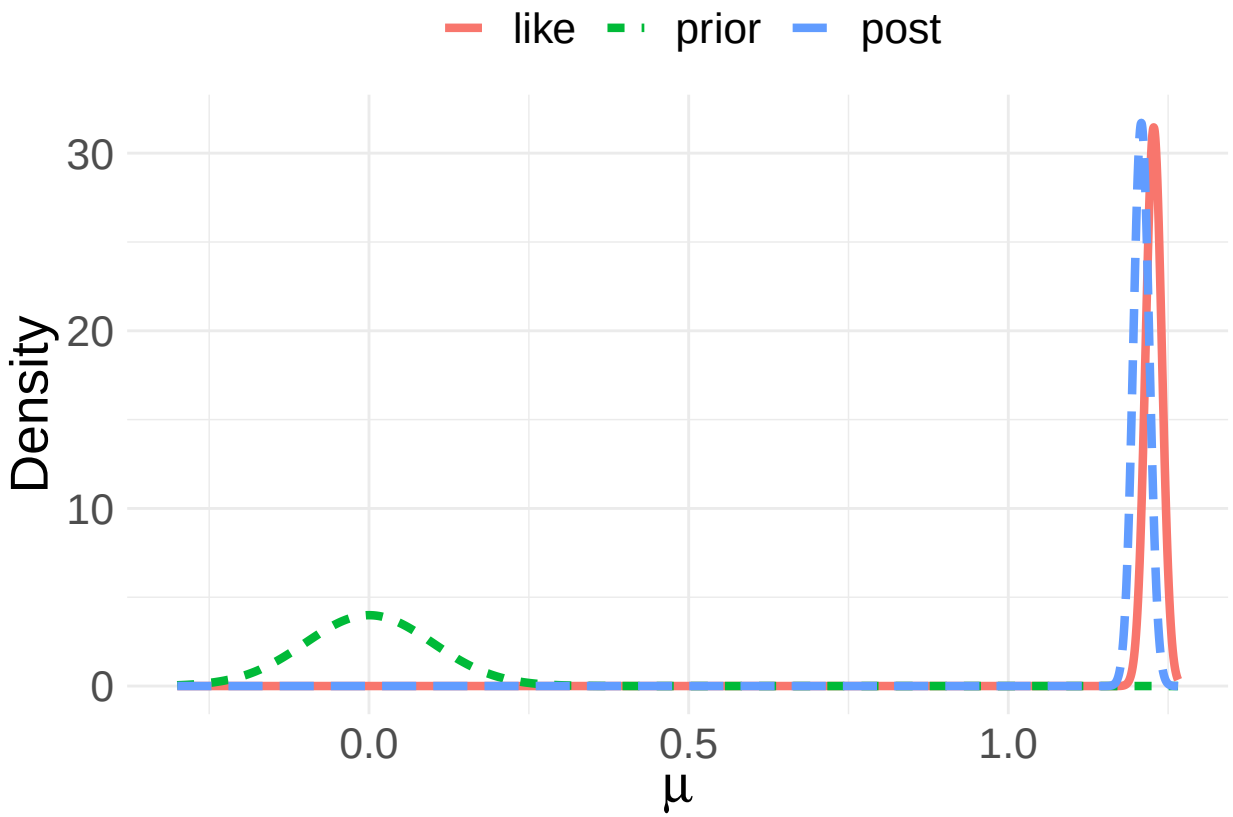
```
prior_dens <- function(x) dnorm(x, mean = mu0 , sd = sigma.mu0)
post_dens <- function(x) dnorm(x, mean = mupost.mean, sd = mupost.sd )
like <- function(x) dnorm(x,  mean = ybar, sd = sd.ybar)

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)
  )
```




```
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)
  )
```



(ii) a point estimate, 95% credible interval, and 80% credible interval.

```
mupost.mean # posterior mean - point estimate
```

```
## [1] 1.20802
```

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

```
## [1] 1.20802
```

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

```
## [1] 1.183359 1.232680
```

```r
qnorm(c(0.1, 0.9), mean = mupost.mean, sd = mupost.sd) # 80% quantile-based CI
```

```
## [1] 1.191895 1.224144
```

Exercise 2c [extra credit 2pts]

Briefly comment on the differences in posteriors between exercises a and b: in which setting is the posterior more data-driven, closer to $\bar{y}$? Is that what you expected?

In exercise b the posterior was more data-driven, and thus closer to $\bar{y}$, which makes sense, because the sample size was much larger. In other words, the increased amount of data meant that more weight was given to the likelihood in the second exercise. Similarly, if one used a much smaller sample size, the results would be more driven by the prior, and thus less data-driven. This is to be expected and hoped for, as if there is more information available (more data), this should influence the results more towards the results of the data (rather than the prior).

Following is a quick calculation with a smaller sample size (45 times smaller than part b, or 9 times smaller than part a), which shows how the prior ends up having a greater influence on the posterior and this is closer to an average of the prior and the likelihood function:

```r
n <- 4635 / 45
```

Fix sigma

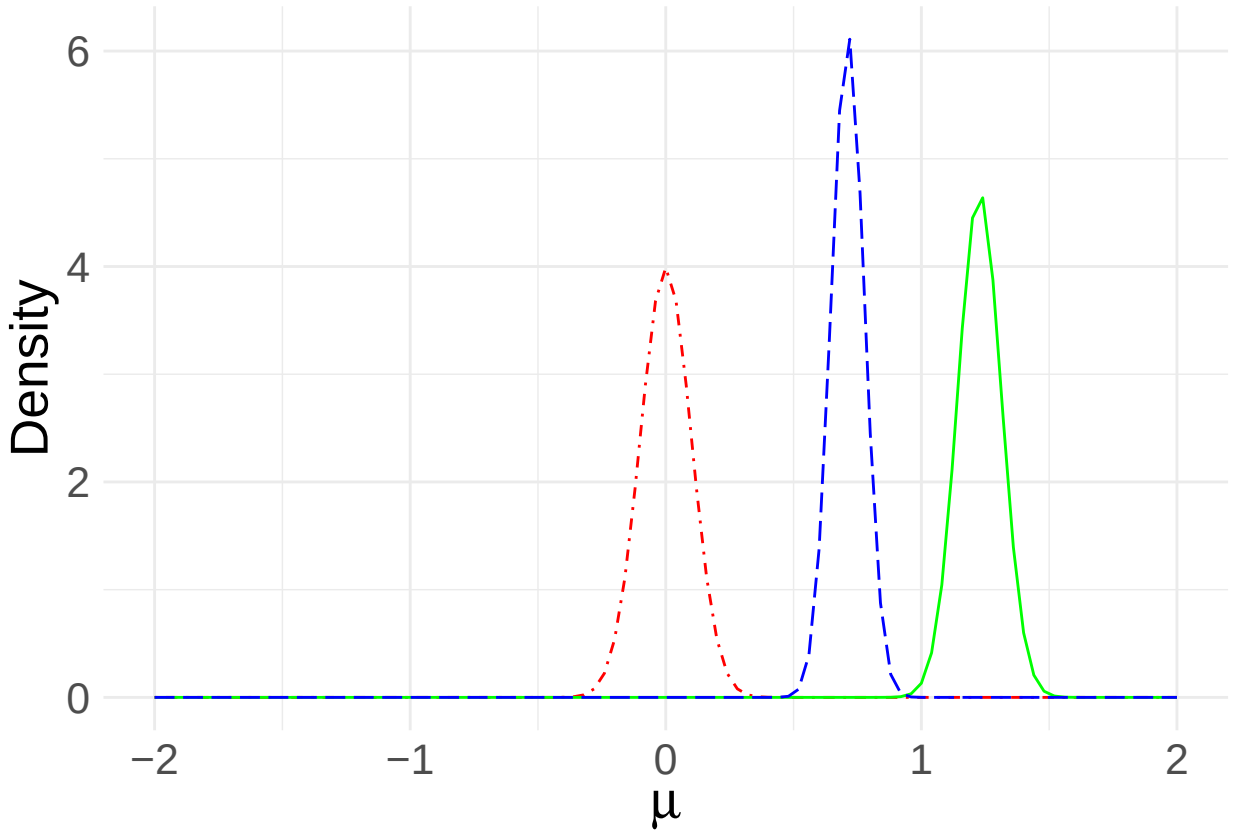
```r
sigma <- sd.y
# sd for ybar follows from sigma
sd.ybar <- sigma/sqrt(n)
```

Then we can obtain posterior mean and variance

```r
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)))
```

(i) one plot with the prior, posterior, and likelihood function;

```r
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)
  )
```




```
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)
  )
```

(ii) a point estimate, 95% credible interval, and 80% credible interval.

```
mupost.mean # posterior mean - point estimate
```

```
## [1] 0.7120405
```

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

```
## [1] 0.7120405
```

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

```
## [1] 0.5850349 0.8390460
```

```
qnorm(c(0.1, 0.9), mean = mupost.mean, sd = mupost.sd) # 80% quantile-based CI
```

```
## [1] 0.628996 0.795085
```

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Prob asymp ♀ has b.c. = 0.8%

If b.c. prob +ve exam = 90%

If not b.c. prob +ve exam = 9%

A = b.c. B = +ve man.

$$P(A|B) = \frac{(0.9)(0.008)}{0.07664} = 0.09394 \approx 9\%$$

(0.0072)

$$P(B) = (0.008)(0.9) + (0.992)(0.07)$$

$$= 0.0072 + 0.06944 =$$

$$= 0.07664$$