Lab 2: Bayesian Inference for Unknown Mean μ

Author:_____Shang Chieh (Jay), Wei_____

Total Grade for Lab 2: /20

Comments (optional)

Template for lab report

Instructions: This is the template you will use to type up your responses to the exercises. To produce a document that you can print out and turn in just click on Knit PDF above. All you need to do to complete the lab is to type up your BRIEF answers and the R code (when necessary) in the spaces provided below.

It is strongly recommended that you knit your document regularly (minimally after answering each exercise) for two reasons.

1. Ensure that there are no errors in your code that would prevent the document from knitting.
2. View the instructions and your answers in a more legible, attractive format.

```
# Any text BOTH preceded by a hashtag AND within the ```{r} ``` code chunk is a comment.
# R indicates a comment by turning the text green in the editor, and brown in the knitted
# document.
# Comments are not treated as a command to be interpreted by the computer.
# They normally (briefly!) describe the purpose of your command or chunk in plain English.
# However, for this class, they will have a different goal, as the text above and below
# each chunk should sufficiently describe the chunk's contents.
# For this class, comments will be used to indicate where your code should go, or to give
# hints for what the code should look like.
```

Bayesian inference summary

We collect a sequence of continuous observations that are assumed identically and independently distributed as Normal(μ, σ), and a normal prior is assigned to the mean parameter μ.

- The sampling model:

$$Y_1, \cdots, Y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma) \tag{1}$$

  When σ (or ϕ) is known, mean μ is the only parameter in the likelihood.

- he prior distribution:

$$\mu \sim \text{Normal}(\mu_0, \sigma_0) \tag{2}$$

- After $Y_1 = y_1, ..., Y_n = y_n$ are observed, the posterior distribution for the mean μ is another normal distribution with mean $\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}$ and precision $\phi_0 + n\phi$ (thus standard deviation $\sqrt{\frac{1}{\phi_0+n\phi}}$):

$$\mu \mid y_1, \cdots, y_n, \sigma \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right). \tag{3}$$

The CE data example

Obtain the CE data sample from Moodle (a .csv file). Below is the sample R script to take log transformation of the `TotalExpLastQ` variable.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
CEsample <- read_csv("CEsample.csv")
```

```
## Rows: 994 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (7): UrbanRural, TotalIncomeLastYear, Race, TotalExpLastQ, log_TotalInco...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
CEsample$LogTotalExpLastQ <- log(CEsample$TotalExpLastQ)
```

Below is the sample R script to obtain the mean `mu_n` and the standard deviation `sd_n` for the posterior distribution for μ. Note the prior choice $\pi(\mu) \sim \text{Normal}(5, 1)$. Also note the use of the known precision for `phi <- 1.25`.

```
mu_0 <- 5
sigma_0 <- 1
phi_0 <- 1/sigma_0^2
ybar <- mean(CEsample$LogTotalExpLastQ)
phi <- 1.25
n <- dim(CEsample)[1]

mu_n <- (phi_0*mu_0+n*ybar*phi)/(phi_0+n*phi)
sd_n <- sqrt(1/(phi_0+n*phi))
```

Bayesian Inference for Unknown Mean μ

Exercise 1: **Assess the statement "the average log total expenditure of a CU is 9 or more". Report on the comparison of the exact solution and approximation by Monte Carlo simulation.** Hint: For the exact solution, use the `pnorm()` function; for approximation by Monte Carlo simulation, use the `rnorm()` function.

```r
1-pnorm(9, mu_n, sd_n, lower.tail = TRUE)
```

```
## [1] 5.662137e-15
```

```r
pnorm(9, mu_n, sd_n, lower.tail = FALSE)
```

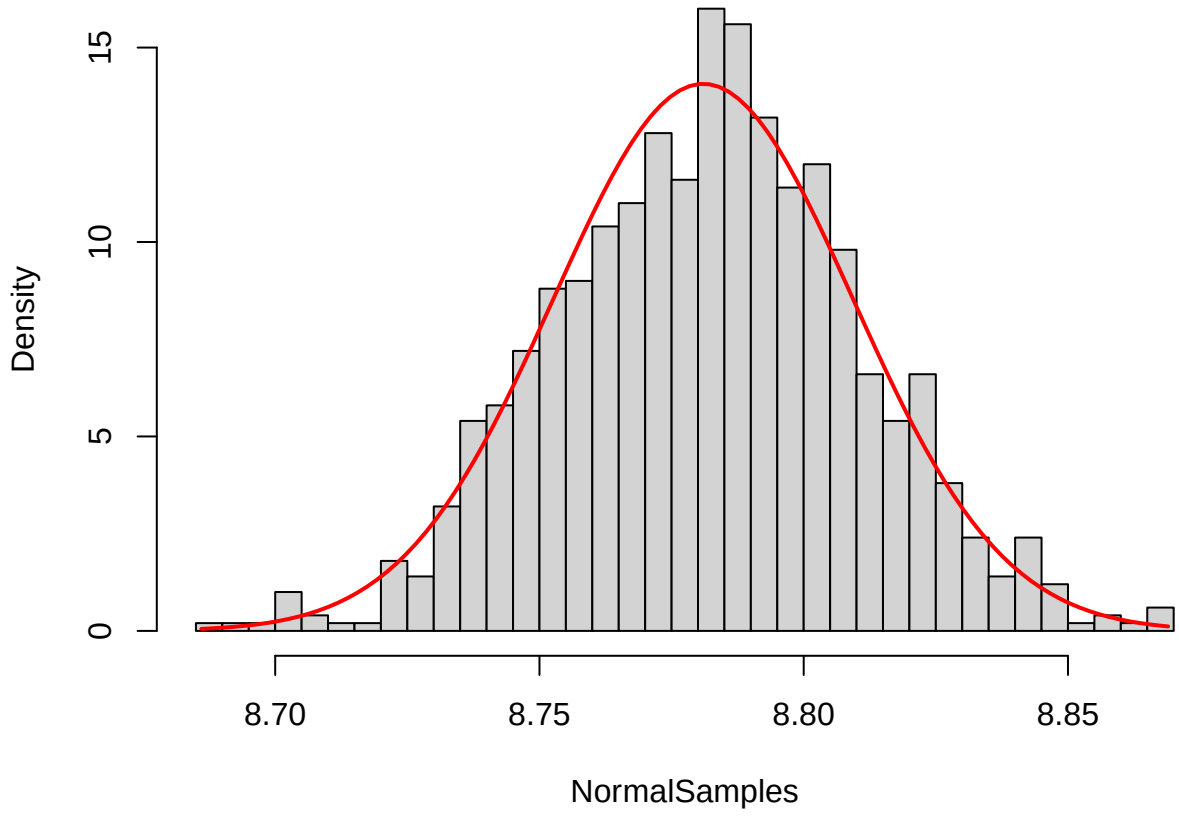
```
## [1] 5.663371e-15
```

```r
# weird, these two should give the same result
```

```r
S <- 1000
NormalSamples <- rnorm(S, mu_n, sd_n)
sum(NormalSamples >=9)/S
```

```
## [1] 0
```

```r
par(mai=c(0.9,0.9,0.1,0.1))
hist(NormalSamples, main=NULL, breaks = 50, prob=TRUE)
x <- seq(min(NormalSamples), max(NormalSamples), length.out=100)
lines(x, dnorm(x, mu_n, sd_n), lwd=2, col="red")
```



Grade for Exercise 1: /4

Comments: As we can see, either from the exact solution or Monte Carlo approximation, the posterior probability is nearly 0. Hence, we reject the claim saying that "the average log total expenditure of a CU is 9 or more."

Exercise 2: **Create a 95% Bayesian credible interval for the parameter μ. Report on the comparison of the exact solution and approximation by Monte Carlo simulation.** Hint: For the exact solution, use the `qnorm()` function; for approximation by Monte Carlo simulation, use the `rnorm()` function.

```
# 95% credible interval
c(qnorm(0.025, mu_n, sd_n), qnorm(0.975, mu_n, sd_n))
```

```
## [1] 8.725398 8.836560
```

```
S <- 1000
NormalSamples <- rnorm(S, mu_n, sd_n)
quantile(NormalSamples, c(0.025,0.975))
```

```
##     2.5%    97.5%
## 8.726277 8.833594
```

Grade for Exercise 2: /4

Comments: One can increase the sample size S, and the Monte Carlo approximation will get closer to the exact solution

Bayesian prediction

One simulates a value from the predictive distribution in two steps: first, one simulates a value of the parameter μ from its posterior distribution, then use this simulated parameter draw to simulate a future observation $\tilde{Y}$ from the sampling distribution. In particular, the following algorithm can be used to simulate a single value from the posterior predictive distribution.

1. Sample a value of μ from its posterior distribution

$$\mu \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \tag{4}$$

2. Sample a new observation $\tilde{Y}$ from the data model (i.e. a prediction)

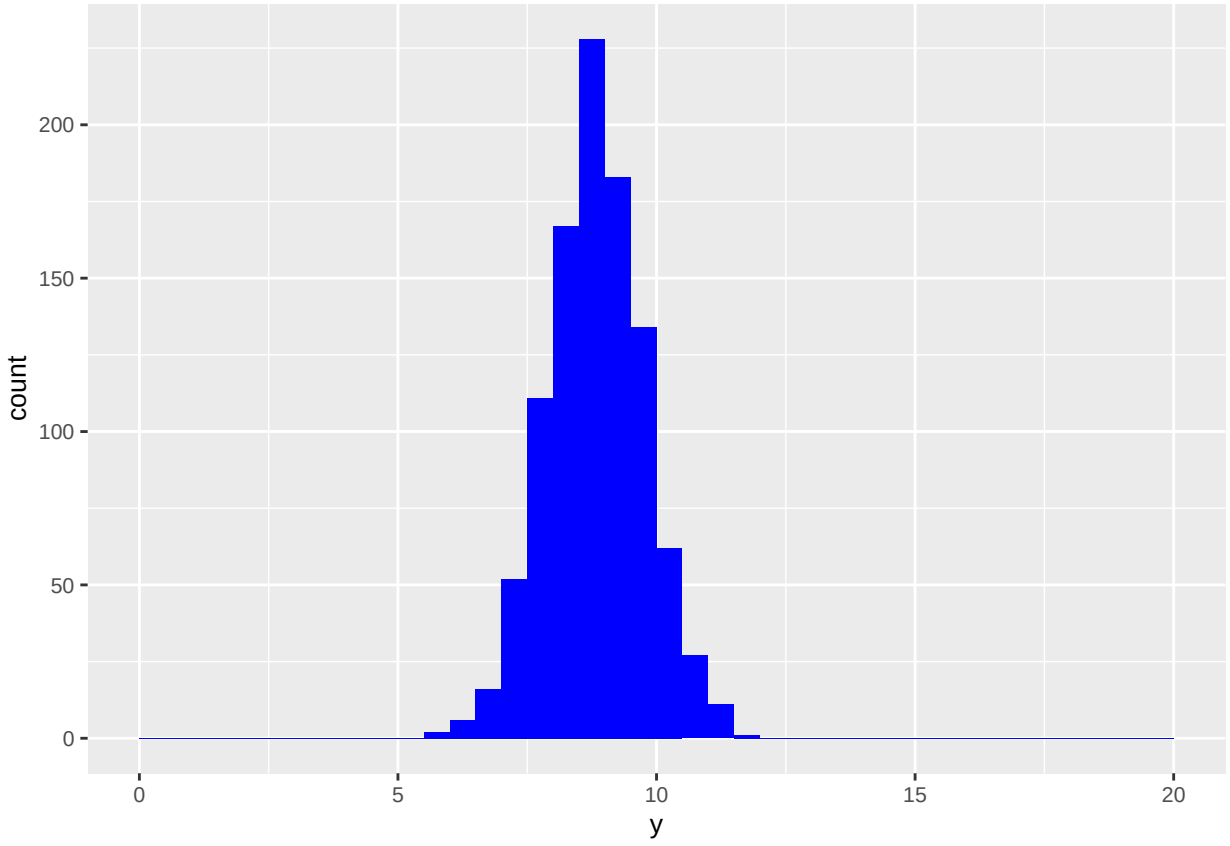
$$\tilde{Y} \sim \text{Normal}(\mu, \sigma). \tag{5}$$

Exercise 3: **Simulate $S = 1000$ predicted values, and make a plot.** Hint: use the `rnorm()` function; use the known `phi <- 1.25` and/or `sigma <- 0.9`.

```r
S <- 1000
pred_mu_sim <- rnorm(S, mu_n, sd_n) #sample mu from posterior

# note that phi=1/sigma^2, hence sigma=1/sqrt(phi)
phi <- 1.25
pred_y_sim <- rnorm(S, pred_mu_sim, 1/sqrt(phi))

y_pred <- data.frame(y=pred_y_sim)
```

```r
# Here is some sample script to create the plot of the predicted log expenditure
# Suppose y_pred is the vector storing the predicted values, and the column name is y

ggplot(data = y_pred, aes(y_pred$y)) +
geom_histogram(breaks = seq(0, 20, by=0.5), fill = "blue") +
xlab("y") + theme(text = element_text(size=10))
```

```
## Warning: Use of `y_pred$y` is discouraged.
## i Use `y` instead.
```



Grade for Exercise 3: /4

Comments: Note that Normal distribution is continuous, so the plot is different from Binomial in Lecture 2

Posterior predictive checking

While Bayesian prediction is focused on simulating one $\tilde{Y}$ from one posterior draw of μ, Bayesian posterior predictive checking is focused on simulating a set of n $\tilde{Y}$'s from one posterior draw of μ, and evaluate the model fitting. In the CE data example, $n = 6208$.

1. Sample a value of μ from its posterior distribution

$$\mu \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \tag{6}$$

2. Sample a set of n new observation $\tilde{Y}_1, \cdots, \tilde{Y}_n$ from the data model (i.e. n predictions)

$$\tilde{Y}_1, \cdots, \tilde{Y}_n \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma). \tag{7}$$

3. To check model fitting, obtain the sample mean $\bar{y}_s$ from the s-th set of $\tilde{Y}_1, \cdots, \tilde{Y}_n$ and compare it against the observed sample mean $\bar{y}$ in the data sample.

Exercise 4: Follow the 3 steps to perform posterior predictive checking.

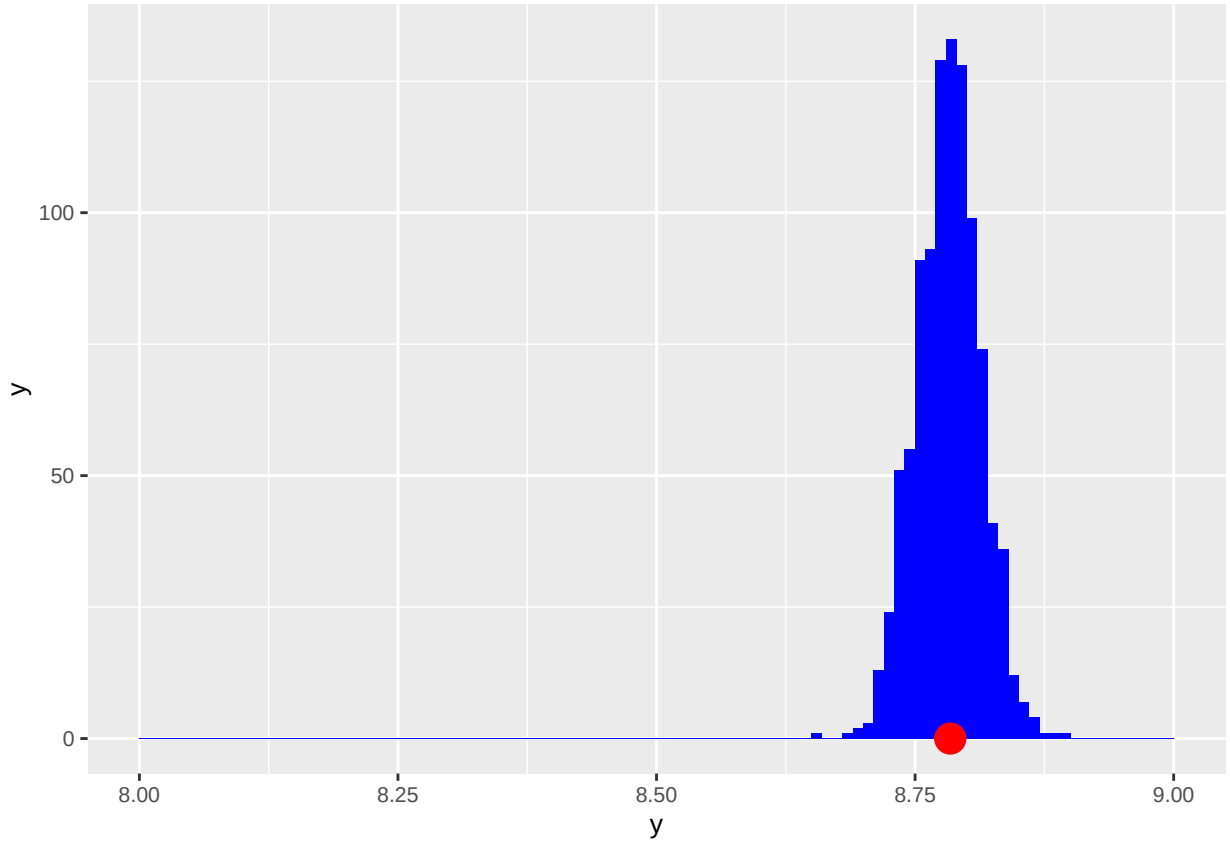
1. Step 1: Simulate $S = 1000$ sets of predicted values, each set contains $n = 6208$ predictions.
2. Step 2: For each set, calculate the sample mean, $\bar{y}_s$.
3. Step 3: Make a plot of $S = 1000$ predicted sample means $\{\bar{y}_s, s = 1, \cdots, S\}$, and compare the sample mean $\bar{y}$ in the CE data sample to the predicted $S = 1000$ sample means. Return $Prob(\bar{y} > \bar{y}_s \mid y)$ and $1 - Prob(\bar{y} > \bar{y}_s \mid y)$ and check the model fitting. Note that if either probability is small, it suggests the model does not describe the data well.

Hint: use the `rnorm()` function; use the known `phi <- 1.25` and/or `sigma <- 0.9`.

```
S <- 1000
n <- 6208
phi <- 1.25
mean_ppc <- as.data.frame(rep(NA, S))
names(mean_ppc)=c("mean")
sample_mean <- mean(CEsample$LogTotalExpLastQ)
for(s in 1:S){
  pred_mu_sim <- rnorm(1, mu_n, sd_n)
  pred_y_sim <- rnorm(n, pred_mu_sim, 1/sqrt(phi))
  mean_ppc[s,] <- mean(pred_y_sim)
}
```

```
# Here is some sample script to create the plot of the predicted sample mean of log expenditure
# Suppose mean_ppc is the vector storing the predicted sample mean values,
# and the column name is mean
# Further suppose the actual sample mean stored as sample_mean

ggplot(data = mean_ppc, aes(mean_ppc$mean)) +
geom_histogram(breaks = seq(8.00, 9.00, by=0.01), fill = "blue") +
annotate("point", x = sample_mean, y = 0, colour = "red", size = 5) +
xlab("y") + theme(text = element_text(size=10))
```

```
## Warning: Use of ‘mean_ppc$mean‘ is discouraged.
## i Use ‘mean‘ instead.
```




```
sum(mean_ppc>sample_mean)/S
```

```
## [1] 0.48
```

```
1-sum(mean_ppc>sample_mean)/S
```

```
## [1] 0.52
```

Grade for Exercise 4: /8

Comments: It seems that the model fits pretty well, the red dot is not at the extreme of the histogram and either probability is large.