ECO518 Assignment 1

Tom Bearpark and Ziqiao Zhang

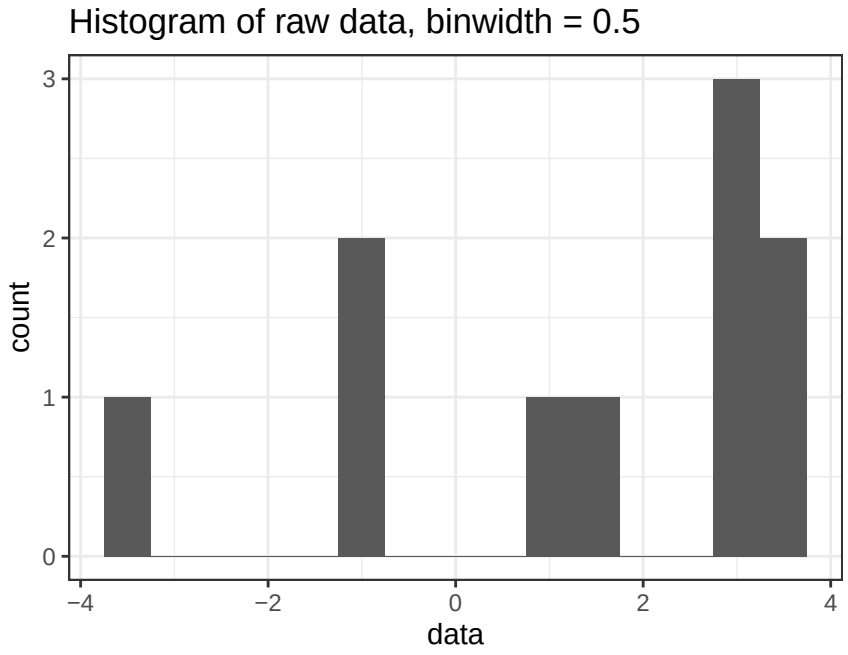
Firstly, we set up our coding environment, and load packages...

```
knitr::opts_chunk$set(fig.width=4.5, fig.height=3.5, fig.align = "center")
rm(list = ls())
library(dplyr, warn.conflicts = FALSE)    # data manipulation, piping
library(ggplot2, warn.conflicts = FALSE) # nice plots
library(purrr, warn.conflicts = FALSE)    # nice loops
library(tidyr, warn.conflicts = FALSE)    # reshaping function
theme_set(theme_bw())                     # ggplot theme
set.seed(123)                             # replicable random numbers
```

Problem 1

Before beginning this problem, we load in the data, and present a histogram of the raw data...

```
data <- c(3.66, 1.00, -0.87, 2.90, -0.80, 3.20, 1.69, -3.53, 3.22, 3.53)
N <- length(data)
mean_raw <- mean(data)
ggplot(data.frame(data = data)) +
  geom_histogram(aes(x = data), binwidth = 0.5) +
  ggtitle("Histogram of raw data, binwidth = 0.5")
```



From the histogram, we can see some evidence of skew, and might also be concerned that our sample is not of the well behaved sort that is amenable to bootstrapping. However, given our very small sample, this evidence is pretty weak.

1. a)

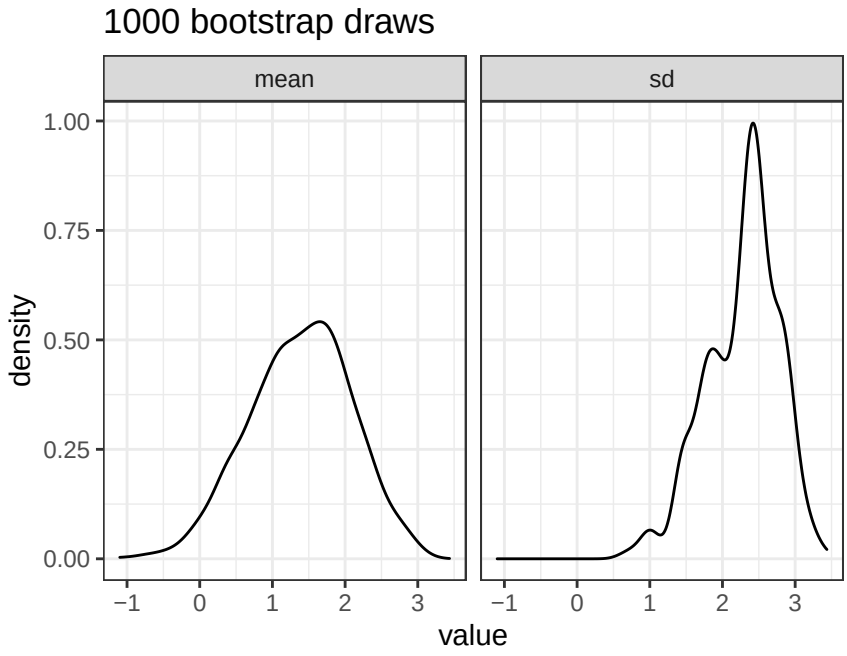
Make 1000 bootstrap draws from this sample of size 10 and use them to construct an estimate of the pdf of the sample mean and the standard deviation of the sample mean.

```
# Define a function for taking draws from the data, returning a
# conveniently formated dataframe
take_draw <- function(i, data, N)
  data.frame(value = sample(data, size = N, replace = TRUE), draw = i)

# Take 1000 draws, save them in a dataframe for convenience.
df <- map_dfr(1:1000, take_draw, data = data, N = N)

# Use this to get estimates of the pdf for the mean and SD
stat_df <- df %>%
  group_by(draw) %>%
  summarise(mean = mean(value), sd = sd(value))
```

1. b)

Plot the estimated pdf, which you can do in R with ***plot****(bkde()) using the KernelSmooth package or with the* ***hist****() function.*

```
# plot using ggplot's "geom_density" function that can estimate the kernel
stat_df %>%
  pivot_longer(cols = -draw, names_to = "statistic") %>%
  ggplot() +
  geom_density(aes(x = value)) +
  facet_wrap(~statistic) +
  ggtitle("1000 bootstrap draws")
```



1. c)

On the assumption that when the mean of the population distribution changes, it changes the distribution by a pure location shift, use your bootstrapped sample to construct a 95% confidence interval for the mean (Should it be "flipped"?)

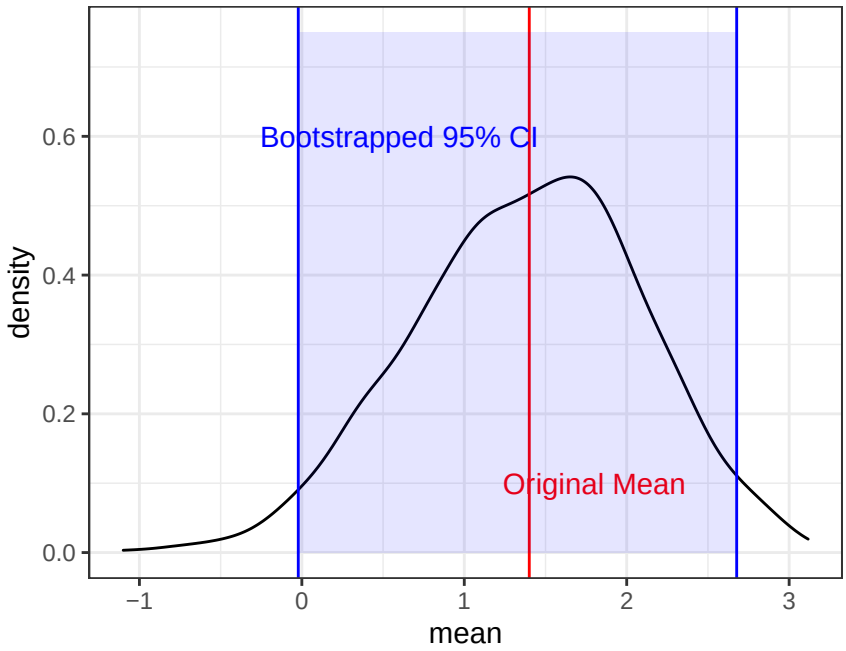
We can construct quantiles of the distribution, by ordering the means of each draw. These will allow us to produce frequentist confidence intervals.

```
CI <- quantile(stat_df$mean, probs = c(0.025, 0.975))
# Show the 95% confidence interval values...
print(CI)
```

```
##    2.5%   97.5%
## -0.0221  2.6771
```

We can visualize these quantiles on a plot. We also present the mean of the original data, so we can analyze the skew.

```
ggplot() +
  geom_density(data = stat_df, aes(x = mean)) +
  geom_vline(xintercept = mean(data), color = "red") +
  geom_text(aes(x = mean(data) + 0.4, y = 0.1, label = "Original Mean"), color = "red") +
  annotate("rect", xmin = CI[1], xmax = CI[2], ymin = 0, ymax = 0.75,
        alpha = .1, fill = "blue") +
  geom_vline(xintercept = CI[1], color = "blue") +
  geom_vline(xintercept = CI[2], color = "blue") +
  geom_text(aes(x = mean(data) - 0.8, y = 0.6,
                label = "Bootstrapped 95% CI"), color = "blue")
```



From the lecture slides, we know that in a location shift model, we can get from confidence intervals to credible sets by 'flipping' the intervals around the mean of the original data. However, in this question, we are asked to find a confidence interval, which does not need to be flipped.

Also, we can see from this plot that the bootstrapped estimates do not appear to be particularly skewed. This suggests that we don't have much need to flip our estimates.

If we do want to be careful, and to say something about the credible set, we can flip the estimates like this:

```
lower_cred <- mean_raw - (CI[2] - mean_raw)
upper_cred <- mean_raw - (CI[1] - mean_raw)
cred_set <- c(lower_cred, upper_cred)
cred_set
```

```
##  97.5%   2.5%
## 0.1229 2.8221
```

As we can see, this is very similar to the version where we didn't flip the interval.

1. d)

The data were actually generated from an equally weighted mixture of two normals, one with mean 0, standard deviation 3, one with mean 3, standard deviation 0.2.

Suppose we knew that the distribution had this form, except for the means of the two components. If the first component has mean − 1.5 and the second mean + 1.5, with unknown, we have a pure location shift model of the data.

Plot the likelihood for this sample as a function of . Does it imply a 95% credible set (under a flat prior) similar to what you got from the bootstrap?

Firstly, lets write down the likelihood.

$$L(Y|\mu) \propto \prod_{i=1}^{10} \left(0.5 * \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_1}{\sigma_1}\right)^2} + 0.5 * \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_2}{\sigma_2}\right)^2} \right)$$

Now, for part (d), we plug in the following parameters...

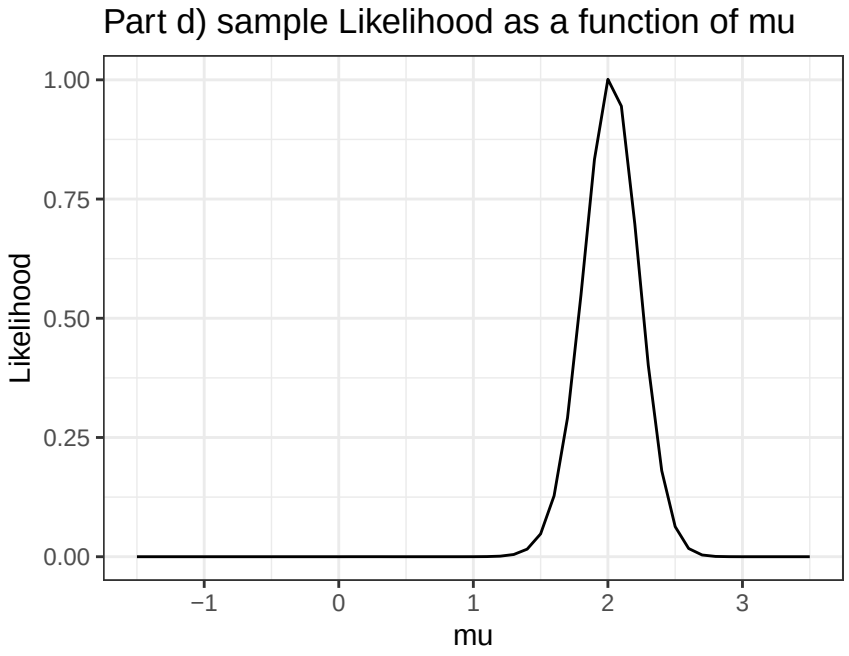
$$\sigma_1 = 3,\ \sigma_2 = 0.2,\ \mu_1 = \mu - 1.5,\ \mu_2 = \mu + 1.5$$

To code this, firstly we write a function that we can use to obtain the likelihood for any required value of mu. This function is general enough to be used in both part d) and part e).

```
l <- function(mu, sigma1, sigma2, X, part){
  if (part == "d"){
    mu_1 <- mu - 1.5
    mu_2 <- mu + 1.5
  } else if (part == "e"){
    mu_1 <- 0
    mu_2 <- 2 * mu
  }
  l <- 1
  for(x_i in X){
    l <- l *
      0.5 *
        (1 / (sigma1 * sqrt(2 * pi))) * exp(-0.5 * ((x_i - mu_1) / sigma1)^2) +
      0.5 *
        (1 / (sigma2 * sqrt(2 * pi))) * exp(-0.5 * ((x_i - mu_2) / sigma2)^2)
  }
  return(l)
}
# initialise a dataframe for plotting
mu_range <- seq(-1.5, 3.5, 0.1)
plot_df <- data.frame(mu = mu_range)
```

Now, we are ready to plot the sample likelihood as a function of mu. To do this, we apply the likelihood function defined above to a range of values of mu for which we want to see the likelihood.

```
plot_df$likelihood_d <- unlist(map(mu_range, l, sigma1 = 3,
                          sigma2 = 0.2, X = data, part = "d" ))
ggplot(plot_df) +
  geom_line(aes(x = mu, y = likelihood_d)) +
  ggtitle("Part d) sample Likelihood as a function of mu") +
  ylab("Likelihood")
```



We assume a flat prior on μ in part (d) and (e), because there is no obvious alternative that is more sensible. Therefore, the posterior probability is proportional to the conditional probability of sample on μ.

Note, the function plotted above is not a posterior probability distribution, since it does not integrate to one. The below code snippet shows that the integral of this function over a reasonable range is not close to one.

```
likelihood_d <- function(mu){
  l(sigma1 = 3, sigma2 = 0.2, X = data, part = "d", mu = mu)
}
const_d <- integrate(likelihood_d, lower = -10, upper = 10)
const_d
```

```
## 0.5181076 with absolute error < 2.9e-05
```

However, we can draw from this posterior distribution using MCMC techniques, in order to find a credible set.

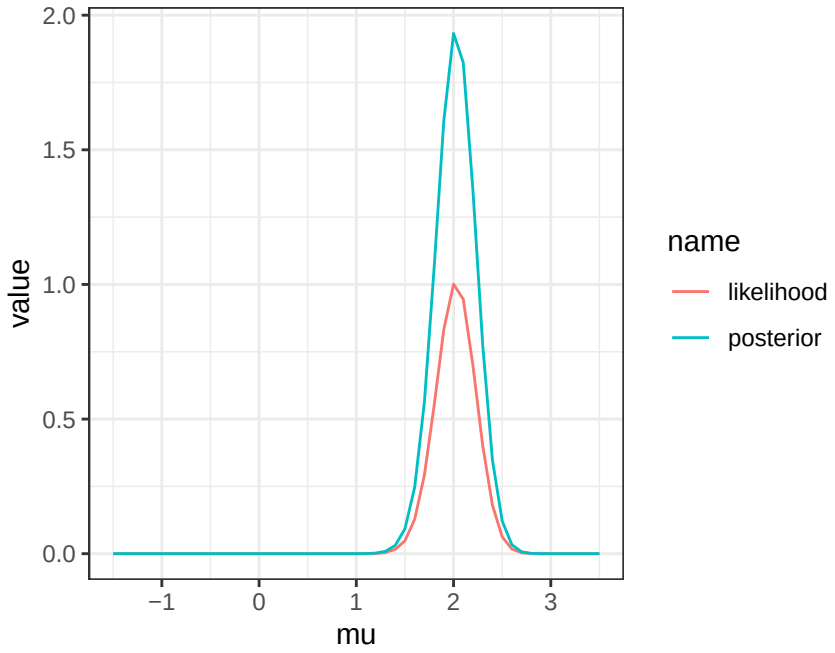
Or alternatively, can we can renormalise the function so that it integrates to one. The below code presents such a function, a provides a plot of the posterior distribution.

```
likelihood_d_norm <- function(mu)  likelihood_d(mu) / const_d$value
integrate(likelihood_d_norm, lower = -10, upper = 10)$value # check that this is 1
```

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

```
plot_df <- plot_df %>%
  mutate(likelihood_d_norm = likelihood_d_norm(mu))

plot_df %>%
  rename(likelihood = likelihood_d, posterior = likelihood_d_norm) %>%
  pivot_longer(cols = -mu) %>%
```

```
  ggplot() +
  geom_line(aes(x = mu, y = value, color = name))
```



Next, we define a function that will find the credible set, given a posterior pdf. To do this, we numerically integrate the function until we find a point on the x axis where the integral is equal to 0.025. We do the same thing starting from the top to get the upper limit of the credible set.

```
get_credible_set <- function(pdf){
  # Get the lower CI
  p <- 0
  lower_ci <- 0
  while(p < 0.025){
    p <- integrate(pdf, lower = -10, upper = lower_ci)
    p <- p$value
    lower_ci <- lower_ci + 0.001
  }

  # Get the upper CI
  p <- 0
  upper_ci <- 5
  while(p < 0.025){
    p <- integrate(pdf, lower = upper_ci, upper = 10)
    p <- p$value
    upper_ci <- upper_ci - 0.001
  }

  print(paste0("Credible set is (", lower_ci, ", ", upper_ci, ")"))
  return(c(lower_ci = lower_ci, upper_ci = upper_ci))
}
```

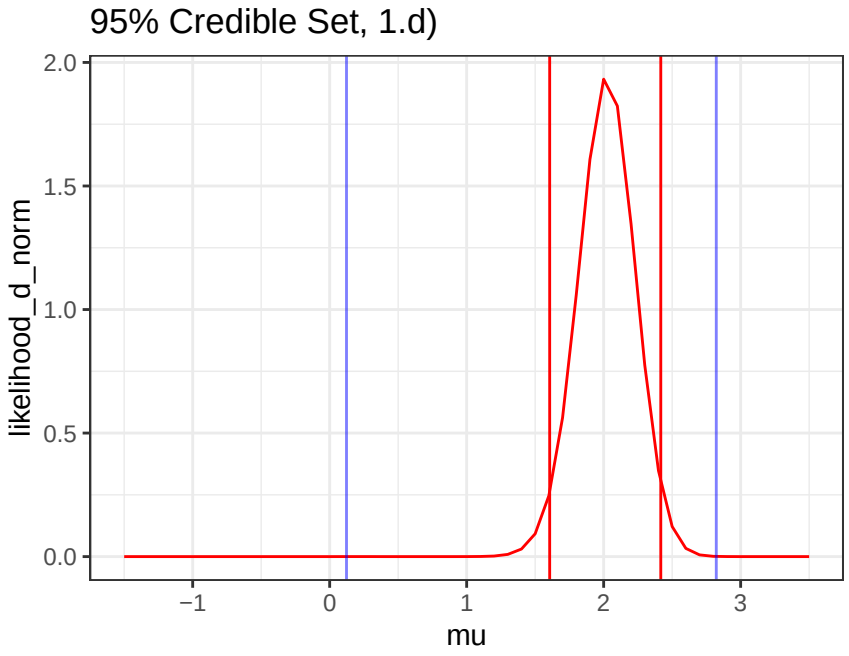
We can run this function for our posterior density found above...

```
credible_set_d <- get_credible_set(likelihood_d_norm)
```

```
## [1] "Credible set is (1.60599999999993, 2.41699999999984)"
```

Finally, lets plot this, and compare it to the bootstrapped estimate. In the below plot, the blue lines represent the bootstrapped credible set found in part c above. The red lines represent the posterior distribution found in part 1d), and the implied 95% credible set calculated above using numerical integration.

```
ggplot(plot_df) +
  geom_line(aes(x = mu, y = likelihood_d_norm), color = "red")  +
  geom_vline(xintercept = credible_set_d, color = "red") +
  ggtitle("95% Credible Set, 1.d)") +
  geom_vline(xintercept = cred_set, color = "blue", alpha = 0.5)
```



We can see that the credible set found in part 1d) is much tighter than that found when we used the bootstrap estimate. Moreover, the credible set is more skewed to the right than confidence set from bootstrap.

Intuition is as follows. First, as the first graph shows, the sample data itself is dense on the right (e.g. multiple occurrences around 3). Therefore, a credible set will be centered to the right of the mean. In contrast, bootstrap takes the sample as population, and makes multiple draws from that. The effect is similar to central limit theorem, making the distribution of bootstrapped sample means looks like a normal, with the 95% CI centered around sample mean.

Moreover, the credible set is tighter, probably because we have more information about the distribution (i.e. the distribution is mixture of normals), which helps narrow the interval. Also, density of normal distribution declines very fast away from the mean, which could also play a role here.

1. e)

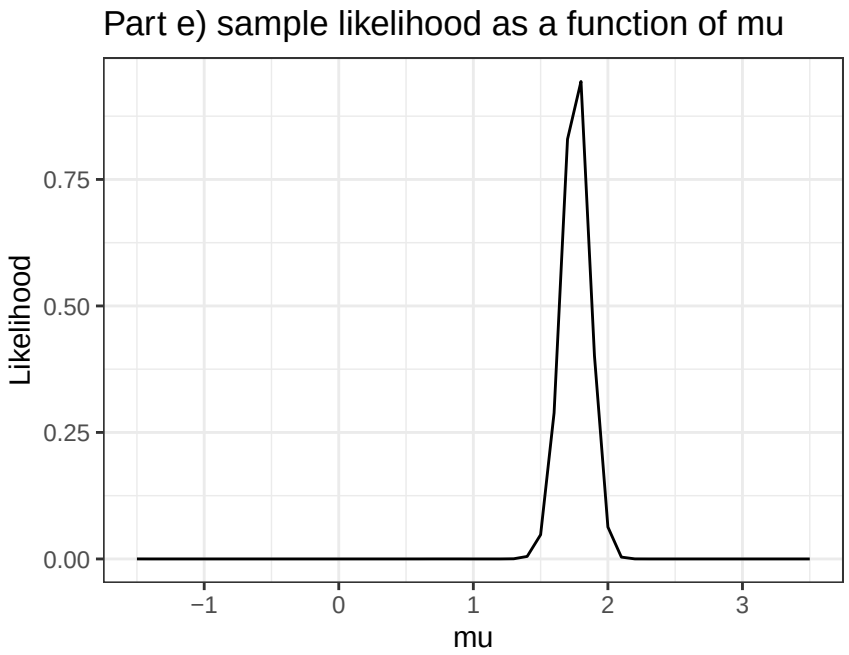
Now suppose instead that we know the first component has mean 0, but don't know the mean of the second component. The mean of the second component is 2 , so that is still the mean of the distribution, but no longer produces a pure location shift. Plot the likelihood and find a 95% credible set for for this case.

For part (e), plug in the following parameters

$$\sigma_1 = 3,\ \sigma_2 = 0.2,\ \mu_1 = 0,\ \mu_2 = 2\mu$$

We can use the function defined in part d) to find our sample likelihood as a function of mu, and plot...

```
plot_df$likelihood_e <- unlist(map(mu_range, l, sigma1 = 3,
                        sigma2 = 0.2, X = data, part = "e"))
ggplot(plot_df) +
  geom_line(aes(x = mu, y = likelihood_e)) +
  ggtitle("Part e) sample likelihood as a function of mu") +
  ylab("Likelihood")
```



Using the same logic as before, we renormalise this function...

```
likielihood_e <- function(mu){
  l(sigma1 = 3, sigma2 = 0.2, X = data, part = "e", mu = mu)
}
const_e <- integrate(likielihood_e, lower = -10, upper = 10)
likielihood_e_norm <- function(mu) likielihood_e(mu) / const_e$value
plot_df <- plot_df %>%
  mutate(likielihood_e_norm = likielihood_e_norm(mu))
```

And then use the function written earlier to calculate a credible set using numerical integration...

```
credible_set_e <- get_credible_set(likielihood_e_norm)
```

```
## [1] "Credible set is (1.55099999999994, 1.95799999999989)"
```

Note, this is an even tighter credible set than that found in part (e). Since we are no longer in the location-shift modeling situation, we do not really know how well the credible set and bootstrap estimate compare. We should be wary of our boostrapped estimate, if we are convinced that this is the underlying data generation process!

Comment: The credible set should be tighter because here only one mean is uncertain. In (d), moving the mean of one process ($\mu + 1.5$) away from the observation can sometimes be compensated by moving the mean of the other process ($\mu - 1.5$) closer to the observation. That effect isn't present in this part, and as a result the credible set is tighter.

Problem 2: The lady tasting tea

Part 1

A lady claims to be able to tell whether she has been served tea with the milk put in the cup before, or after, the tea. She is given a sequence of cups of tea, some of which (or perhaps all, or none of which) have had the milk put in first. She announces her guesses, we find out how many of them were correct. How do we assess the evidence for her abilities?

This is a classic example, probably the simplest and most appealing case for randomization inference. There is no need to appeal to hypothetical repetitions of the lady's test, as in the usual frequentist framework. Here there 2^N possible sequences of milk-first/milk-second, where N is the number of trials. So if she is given, say, 5 cups, there are 32 possibilities. If we choose the sequence of cups she is presented with "at random" from these 32 possibilities, the probability that she gets them all right if she is "just guessing" is 1/32, and that she gets at least 4 out of 5 right is 6/32. This lets us construct an exact, finite-sample test of "H0: she's guessing" at a .03125 or .1875 significance level.

But what is the alternative hypothesis? If it is that her guessing is i.i.d. across cups, with probability p of being right on each cup, we can construct a likelihood function for p and ask what is the flat-prior probability of p > .5 given that she has made 4, or 5, correct guesses. Do that. (The posterior is a Beta distribution.)

Let $Y = \{y_1, y_2, ...y_5\}$ denote the observed sample of guesses, where $y_i = 1$ if the guess is correct, and $y_i = 0$ if the guess is incorrect. Let $n = \sum_i y_i$ denote the total number of correct guesses, and let $N = 5$ be the sample size.

Then, we can write the likelihood function for p as follows:

$$L(Y|p) = \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i} = p^n(1-p)^{N-n}$$

Given we have a flat prior, we can derive the posterior, $q(p|Y)$ as follows:

$$q(p|Y) \propto L(Y|p) = p^n(1-p)^{N-n} \propto Beta(n+1, N-n+1)$$

Now, we can use this to estimate the probability that $p > 0.5$, given she made 4 or five correct guesses. To do this, we just take a lot of draws from the relevant *Beta* distribtuion, and count the number where $p > 0.5$.

Firstly, define a function for doing this simulation.

```
post_prob_greater_than_.5 <- function(N,n){
  draws <- rbeta(10000, n + 1, N-n + 1)
  sum(draws>0.5) / length(draws)
}
N <- 5
```

Now, run it for n = 4...

```
post_prob_greater_than_.5(N, 4)
```

```
## [1] 0.8929
```

Therefore, if she guesses four of the cups correctly, we conclude that there is nearly a 90% chance that she is able to tell the difference.

And for n = 5...

```
post_prob_greater_than_.5(N, 5)
```

```
## [1] 0.9847
```

So, if she guesses all five of the cups correctly, we conclude that there is nearly a 99% chance that she is able to tell the difference.

Part 2

Suppose it happens that every one of the 5 test cups of tea has milk first, and she guesses milk first on every one of them. Does this seem like less strong evidence of her ability than if the cups she had been presented had at least some of each type? How might one justify a claim that the evidence is less strong in this case?

As we have set up the analysis for this question, the fact that all five cups are off the same type does not influence the analysis. And, there is a decent intuitive argument to suggest that it should not, either. Since the alternative hypothesis here is that her choices are iid, unless we have reason to believe that some sets of cups are easier for her to guess than others, there is no reason to discriminate between any sets of cups that she has to guess.

To justify the argument that the evidence is less strong in this case, you could argue that if all the cups are the same type, then it is easier to guess them all correctly. This could be justified due to some appeal to human psychology, or perhaps it is evidence of cheating. Maybe, if all cups are the same, its more likely that the lady has cheated somehow, meaning that she is more likely to get them all right.

To formulate this into our model, we could make the alternative hypothesis conditional on the cups. For example, we could say that if all cups are the same, we require that she gets a higher proportion of them correct in order for us to be convinced that she can truly tell the difference.