Homework 4

PSTAT 115, Spring 2023

Due on June 9, 2023 at 11:59 pm

Installing cmdstanr

```
install.packages("cmdstanr",
                 repos = c("https://mc-stan.org/r-packages/",
                           getOption("repos")))

library(cmdstanr)

install_cmdstan()
```

Problem 1. Goal Scoring in the Women's World Cup

The Chinese Women's soccer team recently won the AFC Women's Asian Cup. Suppose you are interested in studying the Asian Cup performance of this soccer team. Let λ be the be the average number of goals scored by the team. We will analyze λ using the Gamma-Poisson model where data Y_i is the observed number of goals scored in the ith Asian Cup game, i.e. we have $Y_i|\lambda \sim Pois(\lambda)$. The Chinese Women's team scored 4, 7, 3, 2, 3 goals in each one of the matches. We ignore the match against India, which didn't take place due to a Covid outbreak on the Indian team.

A priori, we expect the rate of goal scoring to be $\lambda \sim Gamma(a, b)$. According to a sports analyst, they believe that λ follows a Gamma distribution with $a = 1$ and $b = 0.25$.

1a. (5pts) Compute the theoretical posterior parameters a, b, and also the posterior mean.

```
a = 1
b = 0.25
score <- c(4, 7, 3, 2, 3)
n <- length(score)
post_a <- a + sum(score)
post_b <- b + n
print(paste("a:",post_a))
```

```
## [1] "a: 20"
```

```
print(paste("b:",post_b))
```

```
## [1] "b: 5.25"
```

```r
print(paste("Posterior mean:",post_a/post_b))
```

```
## [1] "Posterior mean: 3.80952380952381"
```

1b. (10pts) Create a new Stan file by selecting "File -> New File -> Stan file" in RStudio and name it `women_cup.stan`, use Rstan to report and estimate the posterior mean of the scoring rate by computing the sample average of all Monte Carlo samples of λ.

```r
stan_model <- cmdstan_model(stan_file=paste(getwd(),"/women_cup.stan", sep=""))
women_cup_fit <- stan_model$sample(data=list(N=n, y=score, alpha=a, beta=b),
                                   refresh=0)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 0.0 seconds.
## Chain 2 finished in 0.0 seconds.
## Chain 3 finished in 0.0 seconds.
## Chain 4 finished in 0.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 0.0 seconds.
## Total execution time: 0.9 seconds.
```

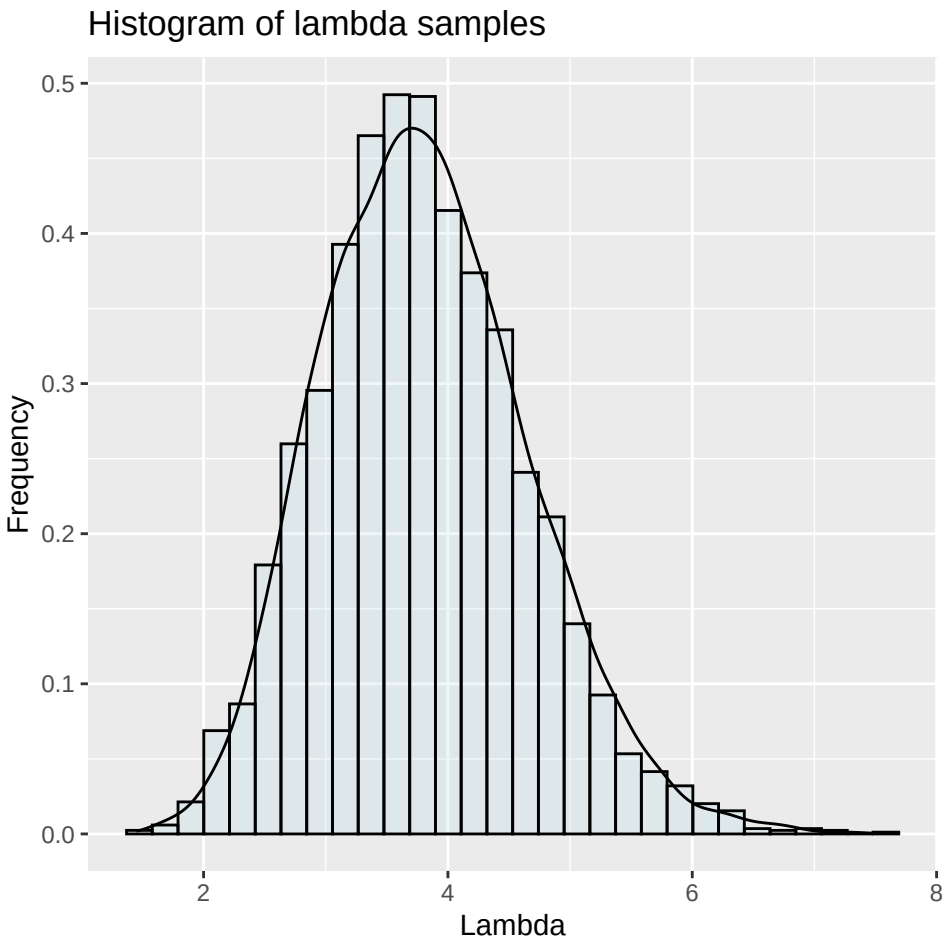
```r
mean_lambda <- women_cup_fit$summary() %>%
    filter(variable == "lambda") %>%
    select("mean")

print(paste("Posterior Mean", mean_lambda))
```

```
## [1] "Posterior Mean 3.783595845"
```

1c. (5pts) Create a histogram of the Monte Carlo samples of λ and add a line showing the theoretical posterior of density of λ. Do the Monte Carlo samples coincide with the theoretical density?

```r
samples <- women_cup_fit$draws(format = "df")
theoretical <- data.frame(lambda = rgamma(4000, post_a, post_b))

# Create histogram
ggplot(samples, aes(x = lambda)) +
    geom_histogram(aes(y = ..density..), color = "black", alpha = .2, fill = "lightblue", bins = 30) +
    geom_density(data = theoretical, aes(x = lambda)) +
    xlab("Lambda") +
    ylab("Frequency") +
    ggtitle("Histogram of lambda samples")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# The Sample density drawn from MC is very close to the theoretical density
```

1d. (10pts) Use the Monte Carlo samples from Stan to compute the mean of the predictive posterior distribution to estimate the distribution of expected goals scored for next game played by the Chinese women's soccer team.

```
print(paste("Mean of posterior predictive:", mean(samples$y_tilde)))
```

```
## [1] "Mean of posterior predictive: 3.81175"
```

Problem 2. Bayesian inference for the normal distribution in Stan.

Create a new Stan file and name it `IQ_model.stan`. We will make some basic modifications to the template example in the default Stan file for this problem. Consider the IQ example in the class slides. Scoring on IQ tests is designed to yield a N(100, 15) distribution for the general population. We observe IQ scores for a sample of n individuals from a particular town, $y_1, \dots y_n \sim N(\mu, \sigma^2)$. Our goal is to estimate the population mean IQ in the town. Assume the $p(\mu, \sigma) = p(\mu \mid \sigma)p(\sigma)$, where $p(\mu \mid \sigma)$ is $N(\mu_0, \sigma/\sqrt{\kappa_0})$ and $p(\sigma)$ is Gamma(a, b). Before you administer the IQ test you believe the town is no different than the rest of the population, so you assume a prior mean for μ of $\mu_0 = 100$, but you aren't to sure about this a priori and so you set $\kappa_0 = 1$ (the effective number of pseudo-observations). Similarly, a priori you assume σ has a mean of 15 (to match the intended standard deviation of the IQ test) and so you decide on setting $a = 15$ and $b = 1$ (remember, the mean of a Gamma is a/b). Assume the following IQ scores are observed:

```
y <- c(70, 85, 111, 111, 115, 120, 123)
n <- length(y)
iq_model <- cmdstan_model(stan_file=paste(getwd(),"/IQ_model.stan", sep=""))
iq_fit <- iq_model$sample(data=list(n=n, y=y),refresh=0)
```

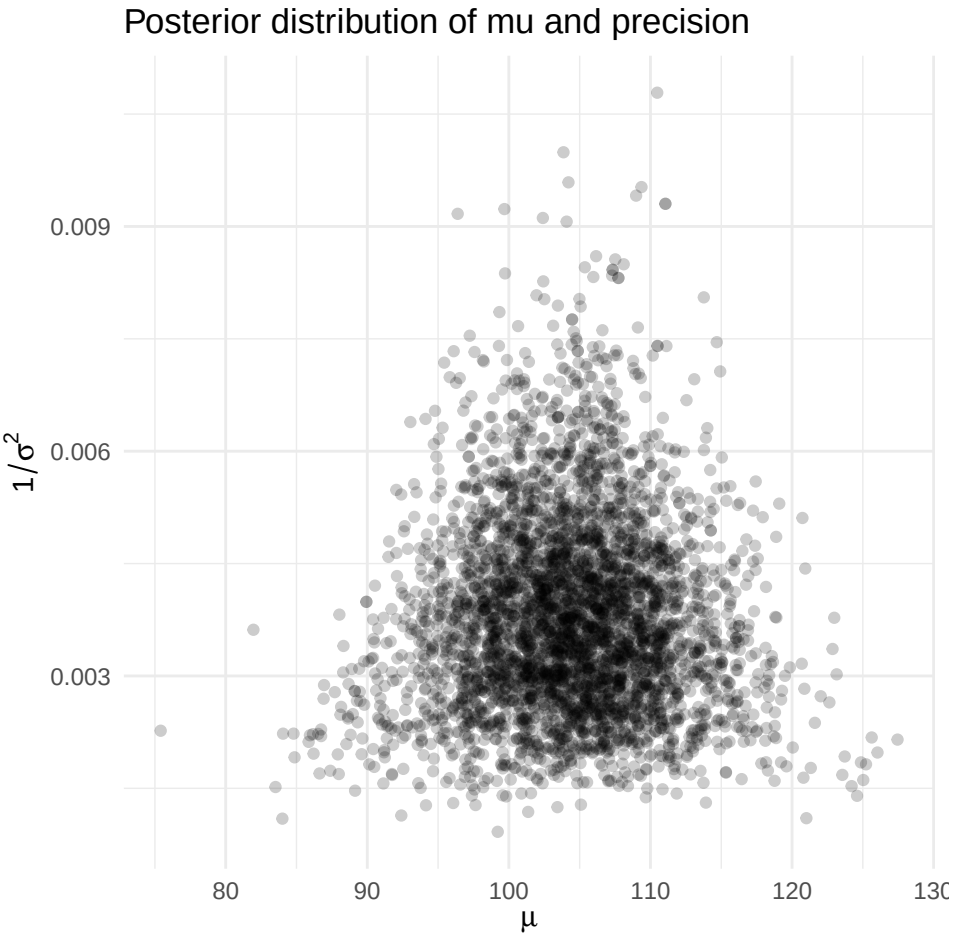
```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 0.0 seconds.
## Chain 2 finished in 0.0 seconds.
## Chain 3 finished in 0.0 seconds.
## Chain 4 finished in 0.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 0.0 seconds.
## Total execution time: 0.7 seconds.
```

2a. (10pts) Make a scatter plot of the posterior distribution of the mean, μ, and the precision, $1/\sigma^2$. Put μ on the x-axis and $1/\sigma^2$ on the y-axis. What is the posterior relationship between μ and $1/\sigma^2$? Why does this make sense? *Hint:* review the lecture notes.

```
iq_samples <- iq_fit$draws(format = "df")
iq_samples$precision = 1/iq_samples$sigma^2
iq_samples
```

```
## # A draws_df: 1000 iterations, 4 chains, and 5 variables
##    lp__  mu sigma mu_gt_100 precision
## 1   3.1 109    17         1    0.0036
## 2   3.0 103    19         1    0.0029
## 3   3.1 104    18         1    0.0030
## 4   2.5  97    18         0    0.0032
## 5   2.0  95    17         0    0.0033
## 6   1.4  93    18         0    0.0031
## 7   1.6  94    20         0    0.0026
## 8   3.1 107    15         1    0.0047
## 9   3.4 103    17         1    0.0037
## 10 -2.3 101    29         1    0.0012
## # ... with 3990 more draws
## # ... hidden reserved variables {'.chain', '.iteration', '.draw'}
```

```
ggplot(iq_samples, aes(x = mu, y = precision)) +
  geom_point(alpha = 0.2) +
  theme_minimal() +
  labs(x = expression(mu), y = expression(1/sigma^2)) +
  ggtitle("Posterior distribution of mu and precision")
```



the inverse of the precision determines the variance of the mu, therefore when the precision is small, the mu spreads more. And when precision is large, the mu spreads less. On the plot, the relationship is illustrated by an uppper triangle shape. As the precision increase, the mu is getting closer

2b. (5pts) You are interested in whether the mean IQ in the town is greater than the mean IQ in the overall population. Use Stan to find the posterior probability that μ is greater than 100.

```
# length(iq_samples$mu[iq_samples$mu > 100])/4000
print(paste("Pr mean greater than 100 (Normal):", mean(iq_samples$mu_gt_100)))
```

```
## [1] "Pr mean greater than 100 (Normal): 0.7755"
```

2c. (15pts) You notice that two of the seven scores are significantly lower than the other five. You think that the normal distribution may not be the most appropriate model, in particular because you believe some people in this town are likely have extreme low and extreme high scores. One solution to this is to use a model that is more robust to these kinds of outliers. The Student's t distribution and the Laplace distribution are two so called "heavy-tailed distribution" which have higher probabilities of outliers (i.e. observations further from the mean). Heavy-tailed distributions are useful in modeling because they are more robust to outliers. Fit the model assuming now that the IQ scores in the town have a Laplace distribution, that is $y_1, \ldots, y_n \sim Laplace(\mu, \sigma)$. Create a copy of the previous stan file, and name it `IQ_laplace_model_1.stan`. *Hint:* In the Stan file you can replace `normal` with `double_exponential` in the model section, another name for the Laplace distribution. Like the normal distribution it has two arguments, μ and σ. Keep the same

prior distribution, $p(\mu, \sigma)$ as used in the normal model. Under the Laplace model, what is the posterior probability that the median IQ in the town is greater than 100? How does this compare to the probability under the normal model? Why does this make sense?

```
iq_model_2 <- cmdstan_model(stan_file=paste(getwd(),"/IQ_model_1.stan", sep=""))
iq_fit_2 <- iq_model_2$sample(data=list(n=n, y=y),refresh=0)
```

```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 0.0 seconds.
## Chain 2 finished in 0.0 seconds.
## Chain 3 finished in 0.0 seconds.
## Chain 4 finished in 0.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 0.0 seconds.
## Total execution time: 0.7 seconds.
```

```
iq_samples_2 <- iq_fit_2$draws(format = "df")
print(paste("Pr medium greater than 100 (Laplace):", mean(iq_samples_2$mu_gt_100)))
```

```
## [1] "Pr medium greater than 100 (Laplace): 0.932"
```

Because of the shape of the Laplace distribution, outliers are getting weighted down, so the two outliers in the data (70, 85) contributes less to the shape of the posterior therefore the mean of the posterior moves to the right

Problem 3. Cows and milk

Farmer John has a huge number of cows. Earlier this year he ran an experiment where he gave 10 cows a special diet that he had heard could make them produce more milk. He recorded the number of liters of milk from these "diet" cows and from 15 "normal" cows during one month. Here is the data:

```
diet_milk = c(651, 679, 374, 601, 401, 609, 767, 709, 704, 679)
normal_milk = c(798, 1139, 529, 609, 553, 743, 151, 544, 488, 555, 257, 692, 678, 675, 538)
```

We want to answer the question: Was the diet any good? Did it result in higher milk production?

Assume the milk production of the cows on the diet is given by a random variable Y_1 and the production of the normal cows is denoted by Y_2. Assume the statistical model is such that

$$
\begin{aligned}
Y_{1,i} \mid \mu_1, \sigma_1 &\sim N(\mu_1, \sigma_1^2), \text{ for } i = 1, \ldots, 10 \\
Y_{2,j} \mid \mu_2, \sigma_2 &\sim N(\mu_2, \sigma_2^2), \text{ for } j = 1, \ldots, 15 \\
\sigma_1 &\sim Unif[0, 1000] \\
\sigma_2 &\sim Unif[0, 1000] \\
\mu_1 \mid \sigma_1 &\sim Unif[0, 2000] \\
\mu_2 \mid \sigma_2 &\sim Unif[0, 2000]
\end{aligned}
$$

3a. (10pts) Write the Stan model into object `cows_string` below.

```
cows_string = "
data {
  int<lower=0> n1;
  int<lower=0> n2;
  array[n1] int<lower=0> y1;
  array[n2] int<lower=0> y2;

}

parameters {
  real<lower=0> mu1;
  real<lower=0> sigma1;
  real<lower=0> mu2;
  real<lower=0> sigma2;
}

model {
  sigma1 ~ uniform(0, 1000);
  mu1 ~ uniform(0,2000);
  sigma2 ~ uniform(0, 1000);
  mu2 ~ uniform(0,2000);

  y1 ~ normal(mu1, sigma1);
  y2 ~ normal(mu2, sigma2);
}

"
```

3b. (5pts) Generate samples from the unknown parameters in the model and save them on an object named `stan_samples`

```
y1 = diet_milk
n1 = length(y1)
y2 = normal_milk
n2 = length(y2)

cows_stan <- write_stan_file(cows_string)
cows_model <- cmdstan_model(stan_file=cows_stan)
cows_fit <- cows_model$sample(data=list(n1=n1, y1=y1, n2=n2, y2=y2),refresh=0)
```

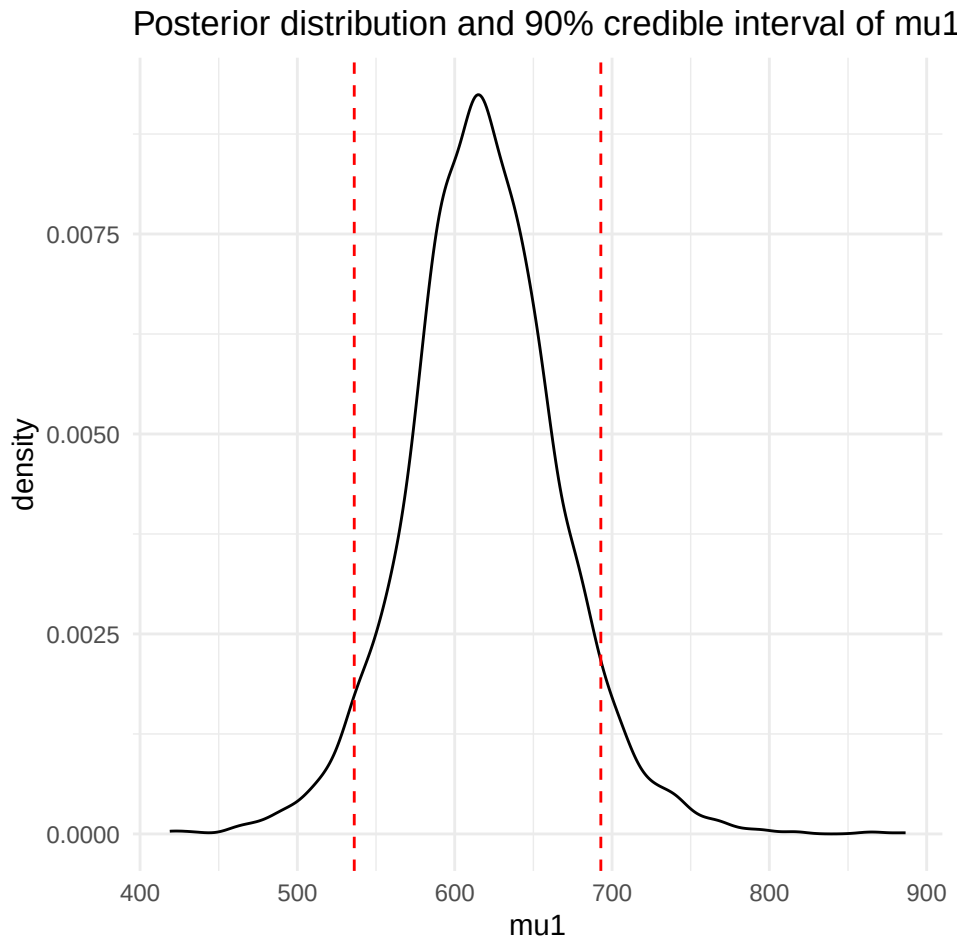
```
## Running MCMC with 4 sequential chains...
##
## Chain 1 finished in 0.0 seconds.
## Chain 2 finished in 0.0 seconds.
## Chain 3 finished in 0.0 seconds.
## Chain 4 finished in 0.0 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 0.0 seconds.
## Total execution time: 0.7 seconds.
```

```
stan_samples = cows_fit$draws(format = "df")
```

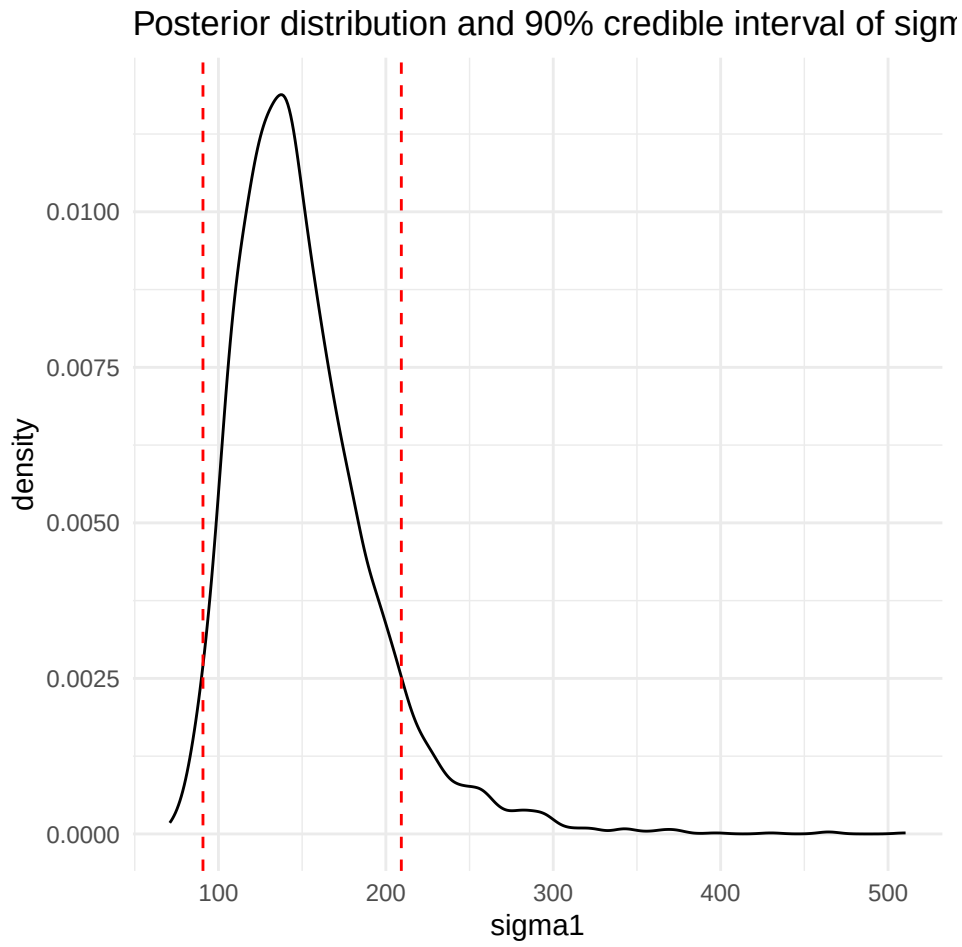
3c. (5pts) Plot the 90% credible interval based on the quantiles of the posterior distribution for each one of the four parameters in the model.

```
library(HDInterval)
mu1_ci = hdi(stan_samples$mu1, credMass = .9)
sigma1_ci = hdi(stan_samples$sigma1, credMass = .9)

mu2_ci = hdi(stan_samples$mu2, credMass = .9)
sigma2_ci = hdi(stan_samples$sigma2, credMass = .9)
```

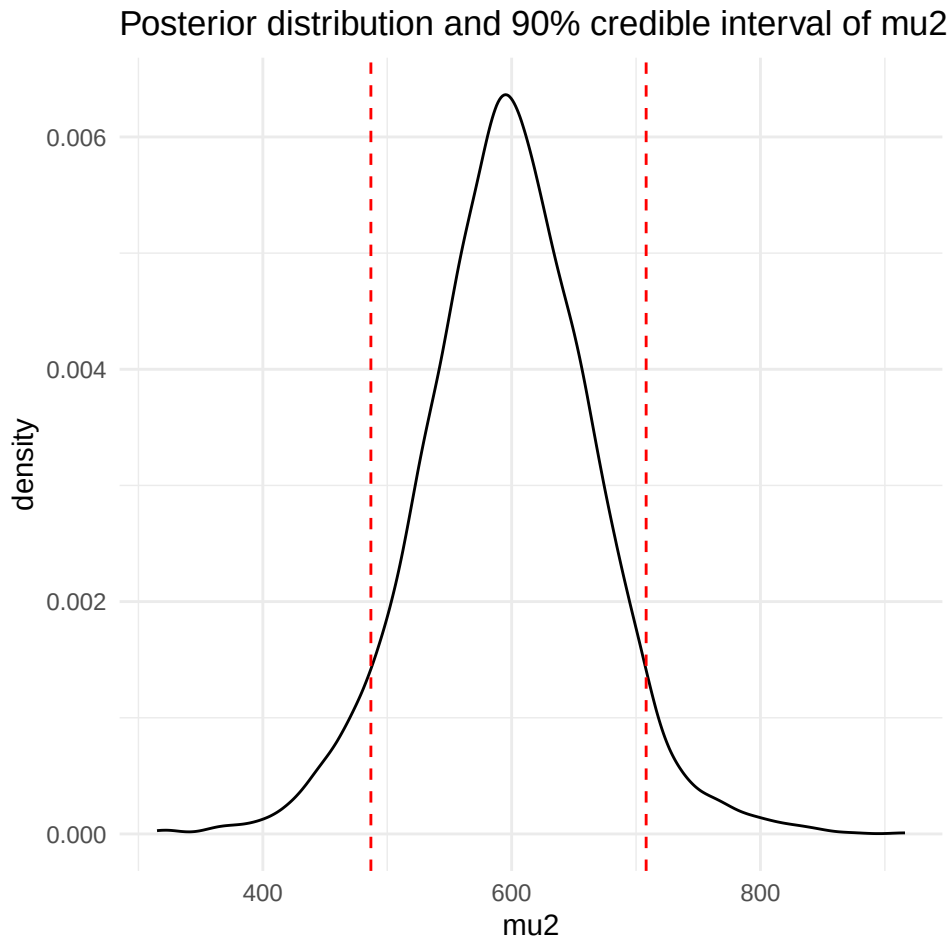
```
plot <- function(data, ci, label) {
    ggplot(data = data.frame(x = data), aes(x = x)) +
        geom_density(col = "black") +
        geom_vline(xintercept = ci[1], color = "red", linetype = "dashed")+
        geom_vline(xintercept = ci[2], color = "red", linetype = "dashed")+
        theme_minimal() +
        xlab(label) +
        ggtitle(paste("Posterior distribution and 90% credible interval of", label))
}

plot(stan_samples$mu1, mu1_ci, "mu1")
```



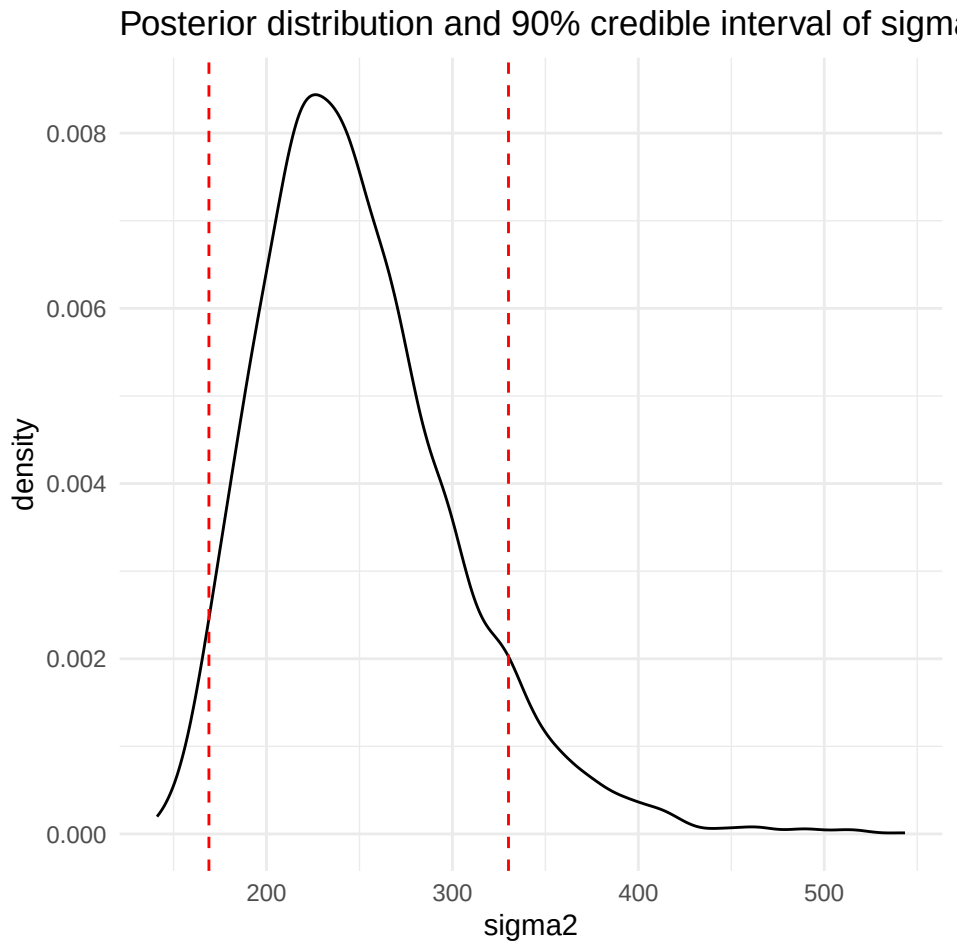

```
plot(stan_samples$sigma1, sigma1_ci, "sigma1")
```




```
plot(stan_samples$mu2, mu2_ci, "mu2")
```



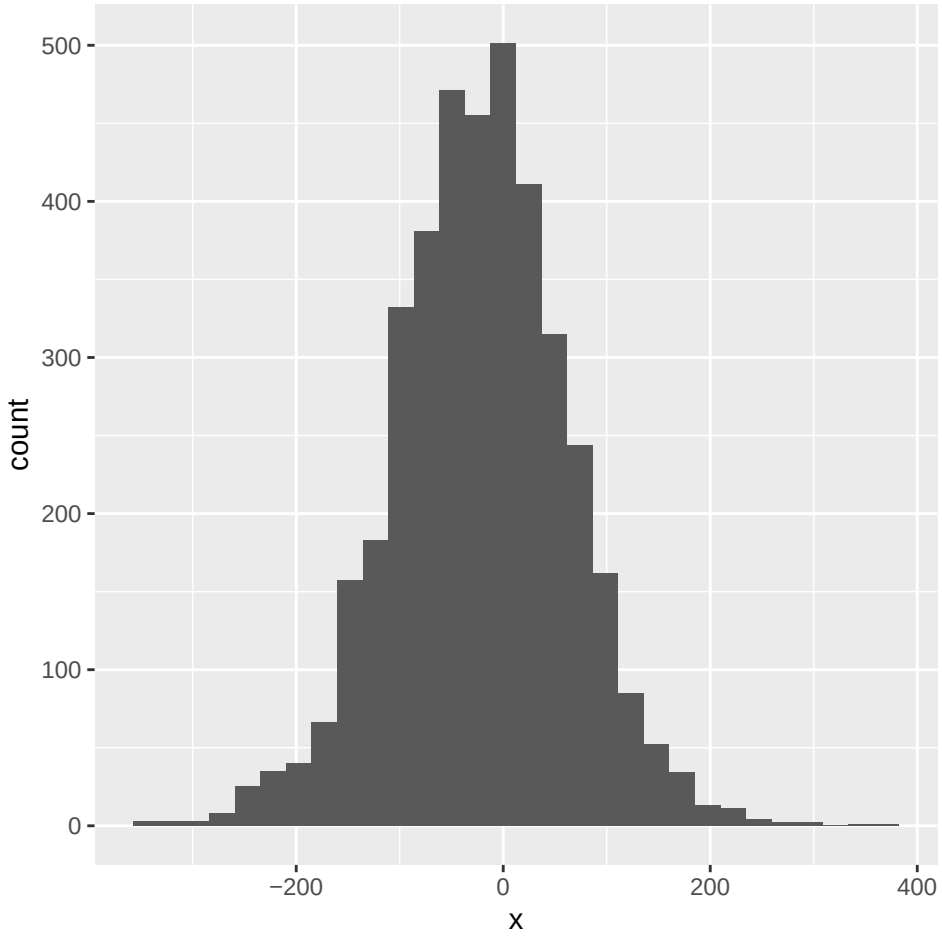

```
plot(stan_samples$sigma2, sigma2_ci, "sigma2")
```



3d. (10pts) Store the posterior samples of the difference $\mu_2 - \mu_1$ on an object named `mu_diff`, plot its histogram and answer the main question: is it likely that the diet is going to make the cows produce more milk on average?

```
mu_diff <- stan_samples$mu2 - stan_samples$mu1
ggplot(data.frame(x = mu_diff), aes(x=x)) +
    geom_histogram()
```

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



not likely since the differences distribution is centered at 0 and has a symmetrical normal shape, indicating a equal likely chance to increase or decrease the milk production.