Homework 4

PSTAT 115, Winter 2023

Due on March 5, 2023 at 11:59 pm

Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

Problem 1. Frequentist Coverage of The Bayesian Posterior Interval.

In the "random facts calibration game" we explored the importance and difficulty of well-calibrated prior distributions by examining the calibration of subjective intervals. Suppose that $y_1, .., y_n$ is an IID sample from a $Normal(\mu, 1)$. We wish to estimate μ.

1a. For Bayesian inference, we will assume the prior distribution $\mu \sim Normal(0, \frac{1}{\kappa_0})$ for all parts below. Remember, from lecture that we can interpret κ_0 as the pseudo-number of prior observations with sample mean $\mu_0 = 0$. State the posterior distribution of μ given $y_1, .., y_n$. Report the lower and upper bounds of the 95% quantile-based posterior credible interval for μ, using the fact that for a normal distribution with standard eviation σ, approximately 95% of the mass is between $\pm 1.96\sigma$.

The posterior distribution would be:

$$exp[\frac{-1}{2}(n + k_0)(\mu - \frac{n\bar{y}}{n + k_0})^2]$$

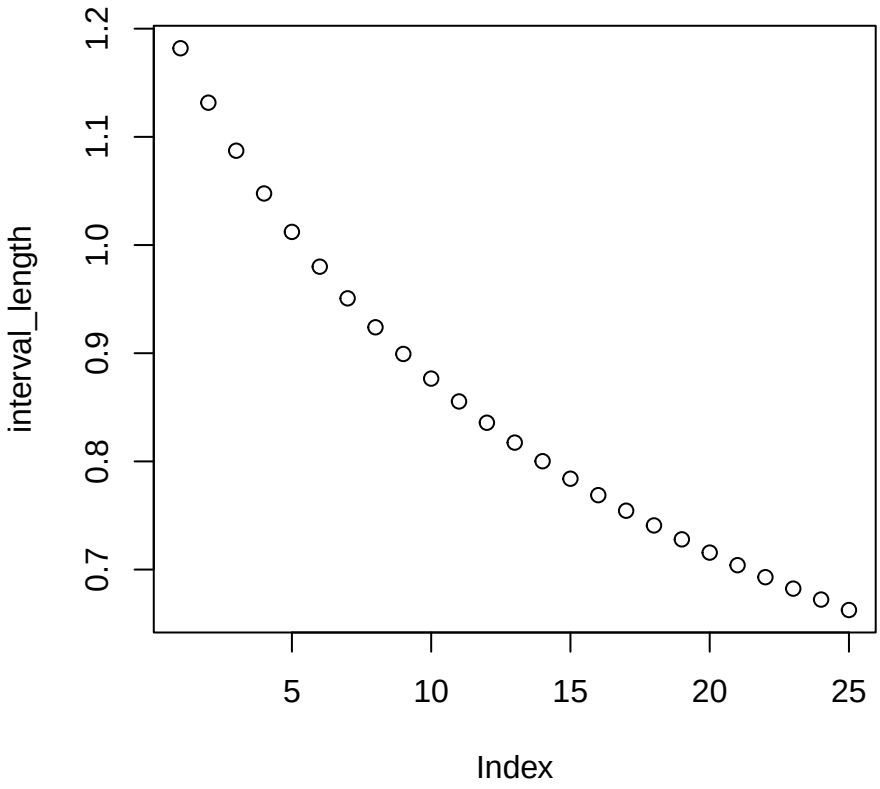
or

$$P(\mu|y_i) \propto \text{Normal}(\frac{n\bar{y}}{k_0 + n}, \frac{1}{k_0 + n})$$

The lower and upper bounds would be $\frac{n\bar{y}}{k_0+n} \pm 1.96\frac{1}{k_0+n}$

1b. Plot the length of the posterior credible interval as a function of κ_0, for $\kappa_0 = 1, 2, ..., 25$ assuming $n = 10$. Report how this prior parameter effects the length of the posterior interval and why this makes intuitive sense.

```
# Use 'interval_length' to store lengths of credible intervals
interval_length <- numeric(25) # YOUR CODE HERE
for(i in 1:25){
  k_0 <- i
  interval_length[i] <- 2 * 1.96 * sqrt(1 / (10 + k_0))
}

## PLOT SOLUTION
plot(interval_length)
```




```
. = ottr::check("tests/q1b.R")
```

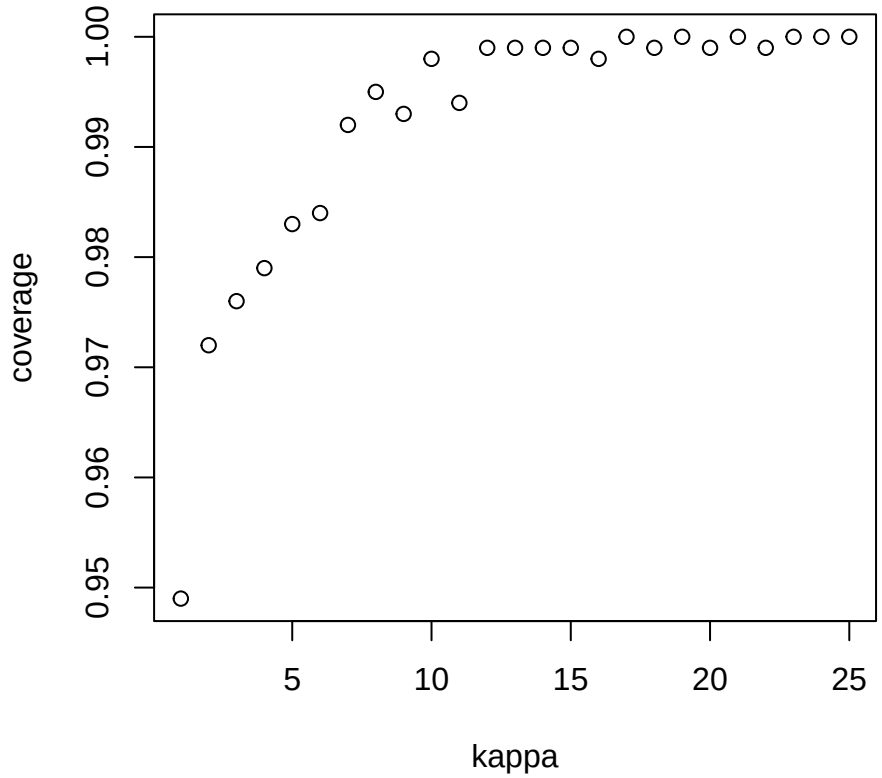
```
##
## All tests passed!
```

This makes intuitive sense because as k_0 increases, the variance and standard deviation decrease. As a result we get a skinnier distribution which means that less length is required for each credible length.

1c. Now we will evaluate the *frequentist coverage* of the posterior credible interval on simulated data. Generate 1000 data sets where the true value of $\mu = 0$ and $n = 10$. For each dataset, compute the posterior 95% interval endpoints (from the previous part) and see if it the interval covers the true value of $\mu = 0$. Compute the frequentist coverage as the fraction of these 1000 posterior 95% credible intervals that contain $\mu = 0$. Do this for each value of $\kappa_0 = 1, 2, ..., 25$. Plot the coverage as a function of κ_0. Store these 25 coverage values in vector called `coverage`.

```
## Fill in the vector called "coverage", which stores the fraction of intervals containing \mu = 0 for
coverage <- numeric(25)

for(i in 1:25){
  k_0 <- i
  count <- 0
  for(j in 1:1000){
    x <- rnorm(10,0, 1)
    lend <- (10 *mean(x) / (k_0 + 10)) - (1.96 * sqrt(1/ (10 + k_0)))
    uend <- (10 *mean(x) / (k_0 + 10)) + (1.96 * sqrt(1/ (10 + k_0)))

    if(lend <= 0 & uend >= 0){
      count <- count + 1
    }

  }
```

```
  coverage[i] <- count / 1000

}

kappa <- c(1:25)

plot(kappa, coverage)
```



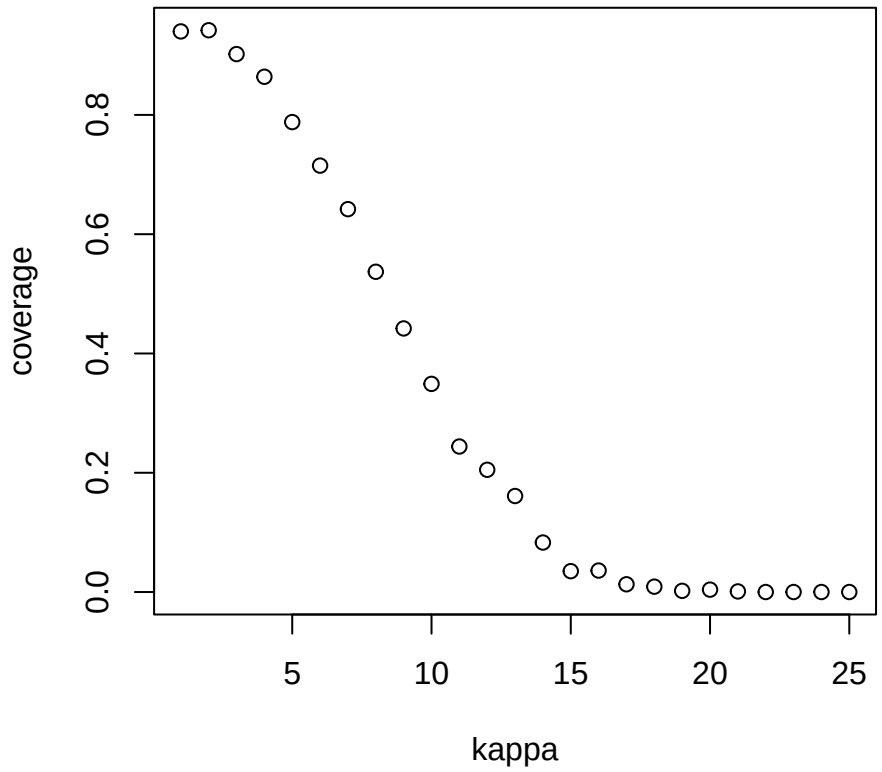

```
. = ottr::check("tests/q1c.R")
```

```
##
## All tests passed!
```

1d. Repeat 1c but now generate data assuming the true $\mu = 1$. Again, store these 25 coverage values in vector called `coverage`.

```
## Fill in the vector called "coverage", which stores the fraction of intervals containing \mu = 1 for
coverage <- numeric(25)

for(i in 1:25){
  k_0 <- i
  count <- 0
  for(j in 1:1000){
    x <- rnorm(10,1, 1)
    lend <-  (10 * mean(x)) / (10 + k_0) - (1.96 * sqrt(1/ (10 + k_0)))
    uend <-  (10 * mean(x)) / (10 + k_0) + (1.96 * sqrt(1/ (10 + k_0)))

    if(lend <= 1 & uend >= 1){
      count <- count + 1
    }
```

```
  }

  coverage[i] <- count / 1000

}

kappa <- c(1:25)

plot(kappa, coverage)
```




```
. = ottr::check("tests/q1d.R")
```

```
##
## All tests passed!
```

1e. Explain the differences between the coverage plots when the true $\mu = 0$ and the true $\mu = 1$. For what values of κ_0 do you see closer to nominal coverage (i.e. 95%)? For what values does your posterior interval tend to overcover (the interval covers the true value more than 95% of the time)? Undercover (the interval covers the true value less than 95% of the time)? Why does this make sense?

When the true μ is zero, the coverage seems to be increasing as a function of kappa. When $\mu = 1$, the function seems to be a decreasing function of kappa. When $\mu = 0$ or $\mu = 1$ the kappa values smaller than 3 appear to have nominal coverage. When $\mu = 0$, for larger kappa values it seems to overcover and when $\mu = 1$ it seems to undercover. These results make sense because our prior data had mean 0, so when our updated data is has mean 1, it will be farther. Our updated data is not similar to our prior, so the coverage will be less.

Problem 2. Goal Scoring in the Women's World Cup

Let's take another look at scoring in soccer. The Chinese Women's soccer team recently won the AFC Women's Asian Cup. Suppose you are interested in studying the World 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 World Cup game, ie. we have $Y_i|\lambda \sim Pois(\lambda)$. *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 $b = 0.25$.

2a. Compute the theoretical posterior parameters a, b, and also the posterior mean.

```
y <- c(4, 7, 3, 2, 3) # Number of goals in each game

post_a <- sum(y) + 1 # YOUR CODE HERE
post_b <- length(y) + 0.25 # YOUR CODE HERE
post_mu <- post_a / post_b # YOUR CODE HERE
```

```
. = ottr::check("tests/q2a.R")
```

```
##
## All tests passed!
```

2b. Create a new Stan file by selecting "Stan file" and name it `women_cup.stan`. Encode the Poisson-Gamma model in Stan. Use `cmdstanr` to report and estimate the posterior mean of the scoring rate by computing the sample average of all Monte Carlo samples of λ.

```
n <- length(y)
## Create "women_cup.stan" yourself and fill in the model
soccer_model <- cmdstan_model("women_cup.stan")

## This fits the model to data y
## All parameter samples are stored in a data frame called "samples"
stan_fit <- soccer_model$sample(data=list(N = n, y = y), refresh=0, show_messages = FALSE)
```

```
## 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: 0.6 seconds.
```

```
samples <- stan_fit$draws(format="df")

## Compute the posterior mean of the lambda samples
post_mean <- mean(samples$lambda)
```

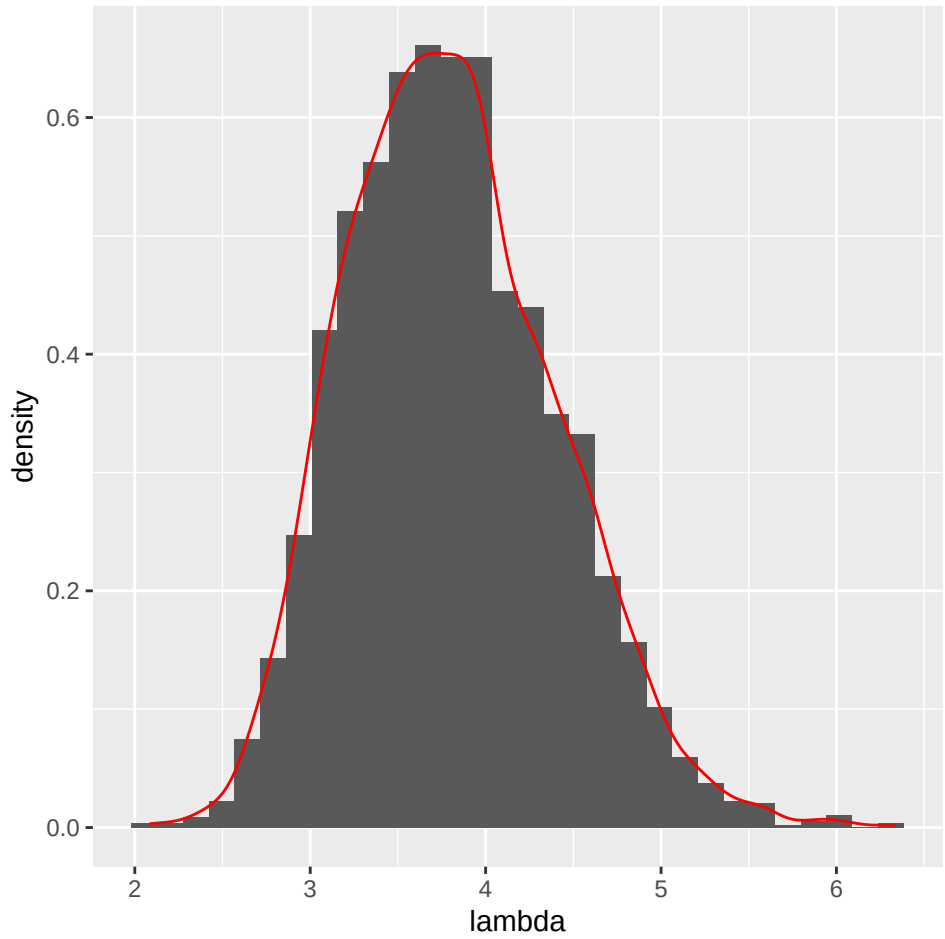
```
. = ottr::check("tests/q2b.R")
```

```
##
## All tests passed!
```

2c. 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?

```
plot <- ggplot(data = samples, aes(x = lambda)) + geom_histogram(aes(y = after_stat(density))) + geom_d

plot
```

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



The Monte Carlo samples do appear to coincide with the theoretical density. Although there are some values far from it, on average it does appear close to it. The shape of both graphs also appear similar and there are not that many outliers.

2d. Use the Monte Carlo samples from Stan to compute the mean of predicative posterior distribution to estimate the distribution of expected goals scored for next game played by the Chinese women's soccer team.

```
pred_mean <- mean(rpois(length(samples$lambda), samples$lambda)) # YOUR CODE HERE
```

```
. = ottr::check("tests/q2d.R")
```

```
##
## All tests passed!
```

Problem 3. 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 used from class. 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, $y_1, \ldots y_n \sim N(\mu, \sigma^2)$. Our goal is to estimate the population mean 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:

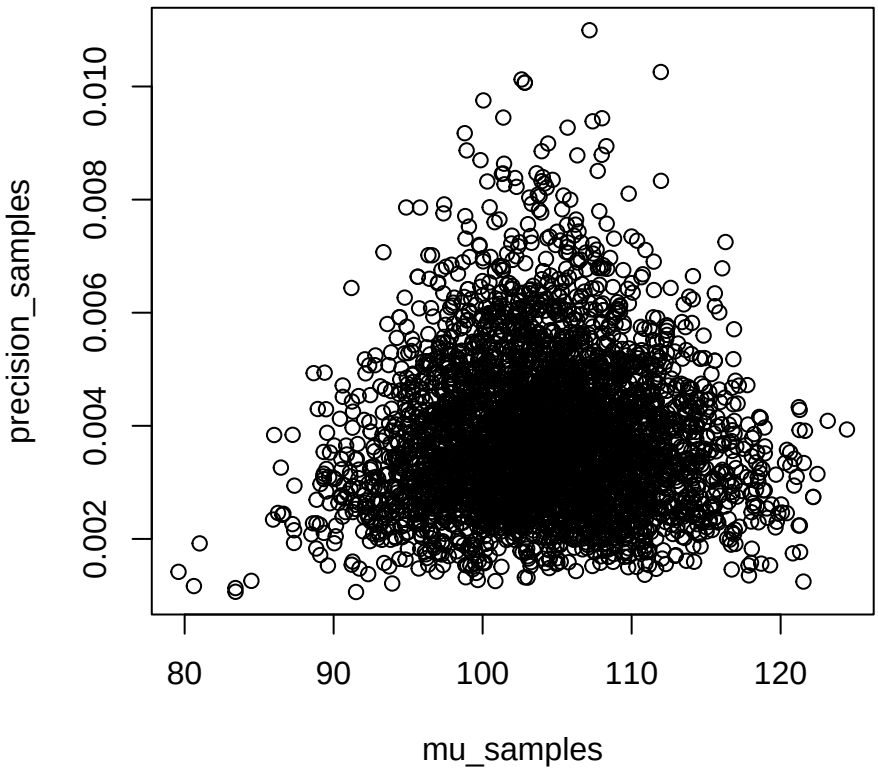
```
y <- c(70, 85, 111, 111, 115, 120, 123)
n <- length(y)
```

3a. 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.

```
normal_stan_model <- cmdstan_model("IQ_model.stan")

# Run rstan and extract the samples
stan_fit <- normal_stan_model$sample(data=list(N = n, y = y), refresh=0, show_messages = FALSE)
```

```
## 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: 0.5 seconds.
```

```
samples <- stan_fit$draws(format="df")

mu_samples <- samples$mu # YOUR CODE HERE
sigma_samples <- samples$sigma # YOUR CODE HERE
precision_samples <- 1 / (samples$sigma^2) # YOUR CODE HERE

## Make the plot
# YOUR CODE HERE
plot(mu_samples, precision_samples)
```




```
. = ottr::check("tests/q3a.R")
```

```
##
## All tests passed!
```

According to our histogram, the precision appears higher when mu is around 105. This makes sense as the precision value is highest when closest to the true mean value. It also appears most dense around that area and further values from 105 tend to have lower precision. In addition, higher precision means lower variance which means we have values closer to the mean. A lower precision means higher variance which leads to values farther from the mean.

3b. 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.

```
mean(mu_samples > 100)
```

```
## [1] 0.75825
```

The probability that μ is greater than 100 is around 0.76

3c. The coefficient of variation, $c_v = \sigma/\mu$ is defined as the standard deviation over the mean. Make a histogram of $p(c_v \mid y)$ from Monte Carlo samples and report the posterior mean and the lower and upper endpoints of the 95% quantile based interval.

```
coefvar <- sigma_samples / mu_samples

mean(coefvar)
```

```
## [1] 0.1631205
```

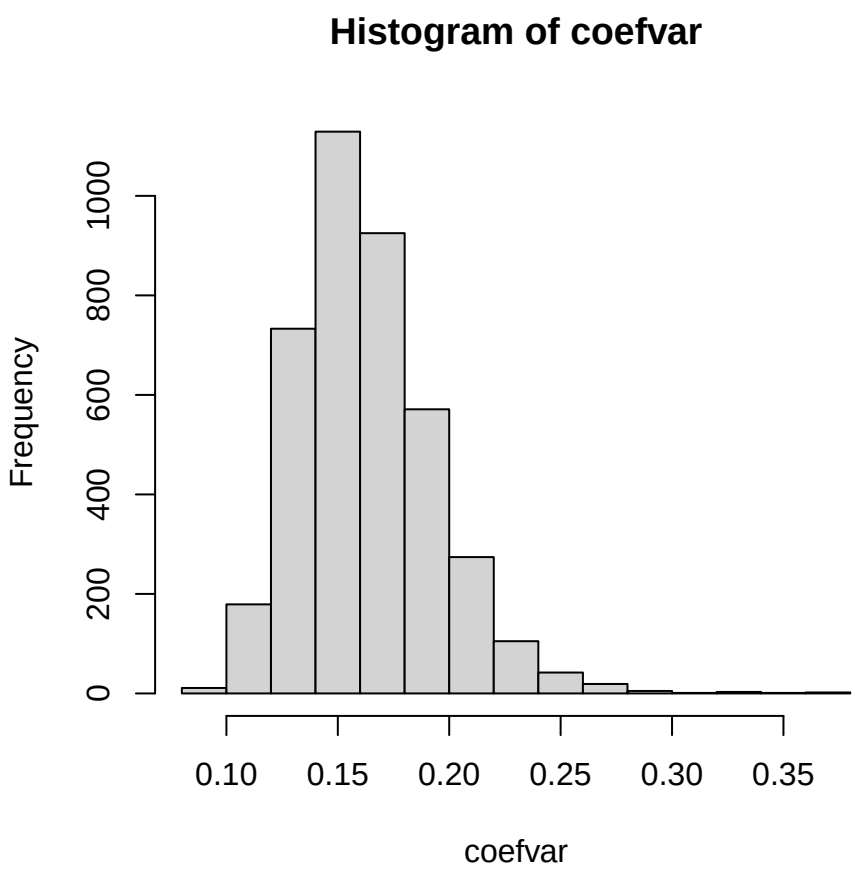
```
quantile(coefvar, 0.025)
```

```
##      2.5%
## 0.1136506
```

```
quantile(coefvar, 0.975)
```

```
##     97.5%
## 0.2344525
```

```
hist(coefvar)
```




```
#just in case the question was asking about mu not the coefficient of variation
mean(mu_samples)
```

```
## [1] 104.247
```

```
quantile(mu_samples, 0.025)
```

```
##     2.5%
## 92.34266
```

```
quantile(mu_samples, 0.975)
```

```
##    97.5%
## 116.6702
```

The posterior mean of the coefficient of variation is 0.16 with the lower quantile value of 0.11 and upper quantile value of 0.23. (The wording of the question is a bit ambiguous so I am not sure if it' asking for the posterior mean and quantiles of μ. I included the code for it in the block and the posterior mean of μ is 104 with a lower quantile of 91 and upper quantile of 116)