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 $\pm 1.96\sigma$.

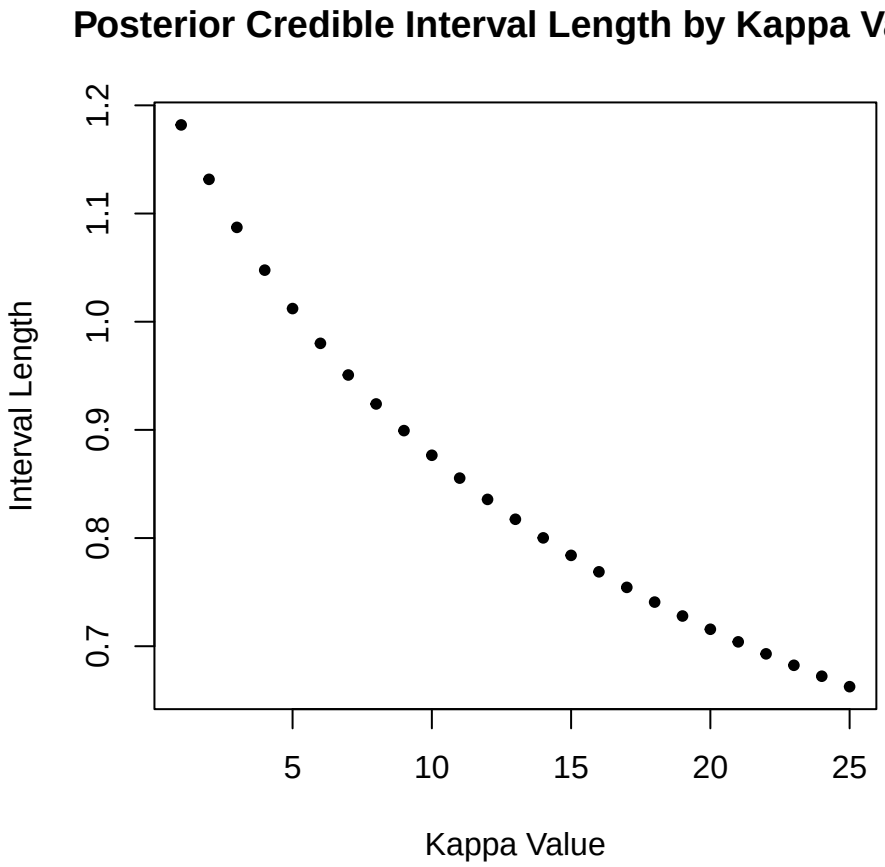
$Var(\mu) = \frac{\sigma^2}{\kappa_0 + n}$ so with $\mu \sim Normal(0, \frac{1}{\kappa_0})$ our posterior variance is now $Var(\mu) = \frac{1}{\kappa_0 + n}$
$\mu = w\bar{y} + (1 - w)\mu_0$ with $\mu_0 = 0$ so our posterior mean is $\mu = \frac{n}{n+k}\bar{y}$

So the Posterior Distribution of μ given $y_1, .., y_n$ is $p(\mu \mid y_1, ..., y_n) \sim N\left(\frac{n}{n+k}\bar{y}, \frac{1}{n+\kappa_0}\right)$

95% quantile-based posterior credible interval for μ is $[\frac{n}{n+k}\bar{y} - 1.96\frac{1}{n+\kappa_0}, \frac{n}{n+k}\bar{y} + 1.96\frac{1}{n+\kappa_0}]$

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.

```
k_0 <- seq(1, 25, length = 25)
n <- 10
post_sd <- sqrt((1/(n + k_0)))

# Use 'interval_length' to store lengths of credible intervals
interval_length <- 2*1.96*post_sd

## PLOT SOLUTION
plot(x = k_0, y = interval_length, main = 'Posterior Credible Interval Length by Kappa Value', xlab = '
```



The more prior observations we have, κ_0, the smaller our variance will be, and thus the smaller our posterior interval will be. This makes sense intuitively because the more prior knowledge we have, the more confident we can be in defining a credible interval, hence a shorter interval.

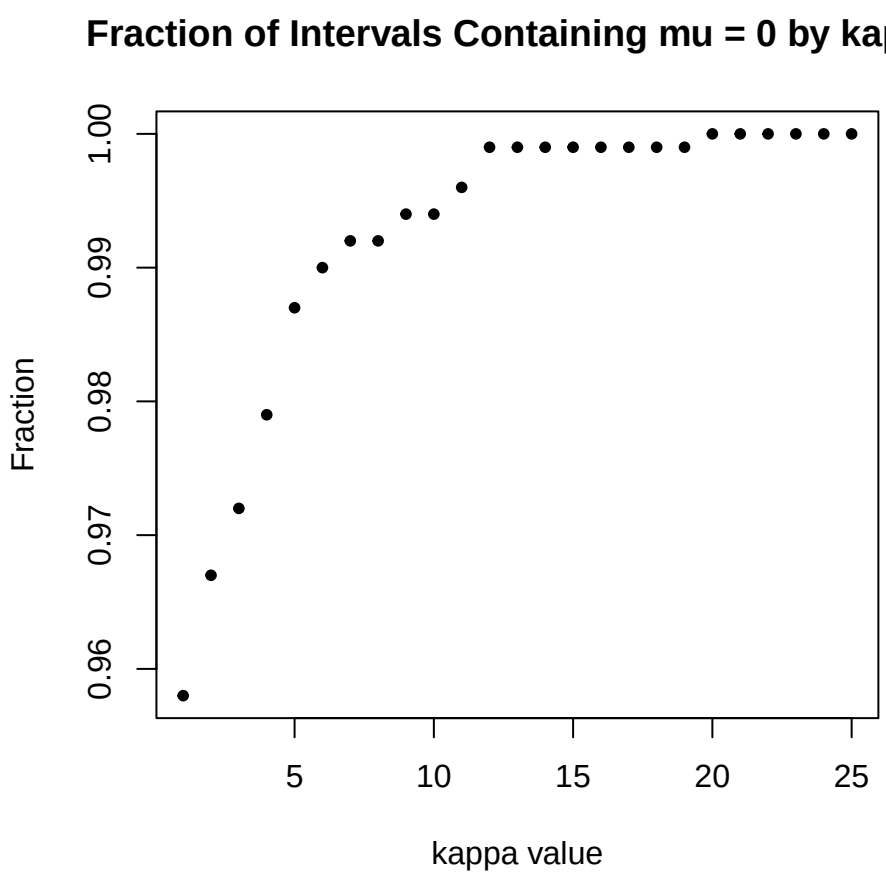
```
. = ottr::check("tests/q1b.R")
```

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

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`.

```
data <- matrix(nrow = 1000, ncol = 10)

for (i in 1:1000){
  data[i,1:10] <- rnorm(10, 0, 1)
}

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

for (i in k_0){
  post_sd <- sqrt((1/(n + k_0[i])))
```

```
  for (j in 1:1000){
    y_bar <- mean(data[j,])
    post_mu <- (n / (n + k_0[i]))*y_bar
    lower <- post_mu - 1.96*post_sd
    upper <- post_mu + 1.96*post_sd
    if (lower <= 0 & upper >= 0){
      coverage[i] <- coverage[i] + 1
    }
  }
}
coverage <- coverage / 1000

# Plot
plot(x = k_0, y = coverage, main = 'Fraction of Intervals Containing mu = 0 by kappa', xlab = 'kappa va
```



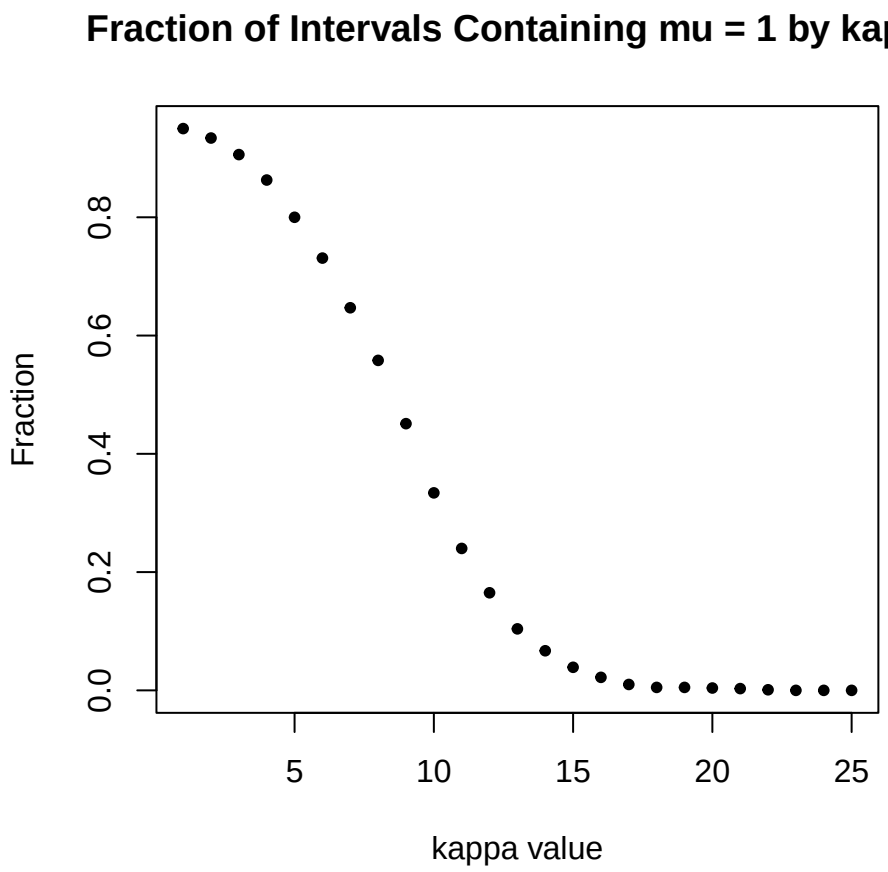

```
. = ottr::check("tests/q1c.R")
```

```
##
## 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`.

```
data2 <- matrix(nrow = 1000, ncol = 10)
for (i in 1:1000){
  data2[i,1:10] <- rnorm(10, 1, 1)
}
```

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

for (i in k_0){
  post_sd <- sqrt((1/(n + k_0[i])))

  for (j in 1:1000){
    y_bar <- mean(data2[j,])
    post_mu <- (n / (n + k_0[i]))*y_bar
    lower <- post_mu - 1.96*post_sd
    upper <- post_mu + 1.96*post_sd
    if (lower <= 1 & upper >= 1){
      coverage[i] <- coverage[i] + 1
    }
  }
}
coverage <- coverage / 1000

# Plot for mu = 1
plot(x = k_0, y = coverage, main = 'Fraction of Intervals Containing mu = 1 by kappa', xlab = 'kappa va
```




```r
. = 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 $\mu = 0$ the coverage mu is very high, while when true $\mu = 1$ the coverage of mu is far worse and for some kappa values there is no coverage at all.

When $\mu = 0$ there is nominal coverage from $\kappa_0 = 1$ but when $\kappa_0 >= 2$ roughly, there begins to be over coverage. Additionally when $\mu = 0$ there is no under coverage. This makes sense because when κ_0 is higher, there is less variance, so $\mu = 0$ is captured within the interval much more.
When $\mu = 1$ there is nominal coverage for $\kappa_0 = 1$ and $\kappa_0 = 2$, but for larger κ_0 values the coverage significantly decreases and we have under coverage. This makes sense because our prior is set for $\mu = 0$, so with $\mu = 1$ and high variance, $\mu = 1$ is still captured, but when variance decrease $\mu = 1$ is no longer captured because the interval is centered around $\mu = 0$.

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
N <- length(y)

post_a <- sum(y) + 1
post_b <- length(y) + 0.25
post_mu <- post_a / post_b
```

```
. = 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.

```
## 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(Y = y, N = N), 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")

## Compute the posterior mean of the lambda samples
post_mean <- stan_fit$summary()[2,2]
post_mean
```

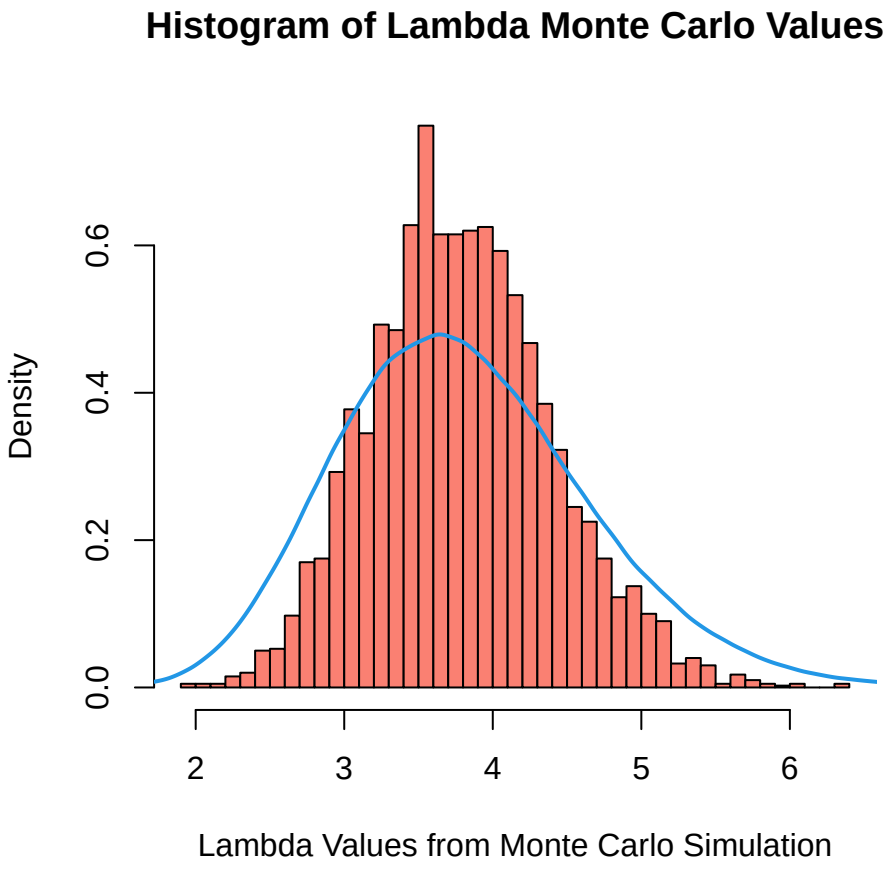
```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1  3.80
```

```
. = 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?

```
points <- rgamma(1000000, 20, 5.25) # Computing theoretical posterior density of lambda

hist(stan_fit$draws('lambda'), breaks = 40, main = 'Histogram of Lambda Monte Carlo Values', xlab = 'La
lines(density(points), col = 4, lwd = 2)
```



The Monte Carlo Samples do coincide with the theoretical posterior density, we see that the peak of the density curve is very close to the peak of the histogram. Furthermore, the rest of the data from the histogram

is distributed similarly to the theoretical posterior density.

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))
pred_mean
```

```
## [1] 3.7685
```

```
. = 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, $y_1, \dots y_n \sim N(\mu, \sigma^2)$. 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:

```
y3 <- c(70, 85, 111, 111, 115, 120, 123)
n3 <- length(y3)

post_a3 <- sum(y3) + 15
post_b3 <- length(y3) + 1
post_mu3 <- post_a3 / post_b3
```

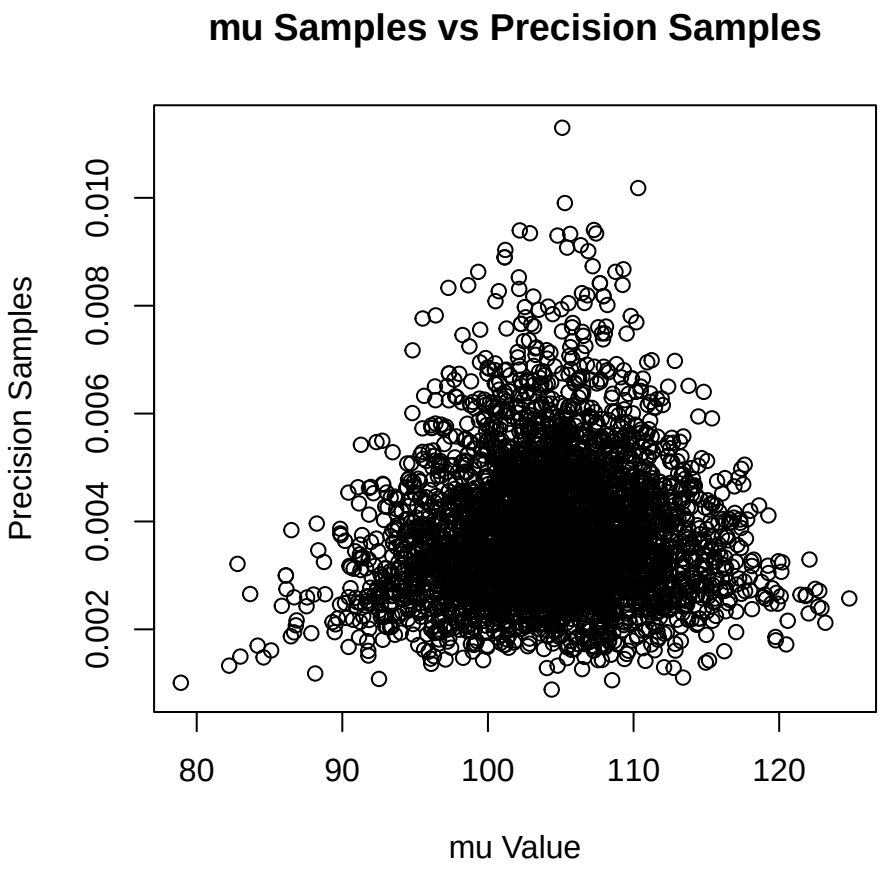
3a. Make a scatter plot of the posterior distribution of the mean, μ, and the precision, $1/\sigma^2$. Put μ on the x-axis and $1/\sigma^2$ on the y-axis. What is the posterior relationship between μ and $1/\sigma^2$? Why does this make sense? *Hint:* review the lecture notes.

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

# Run rstan and extract the samples
stan_fit3 <- normal_stan_model$sample(data=list(Y = y3, N = n3), 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.
```

```
mu_samples <- stan_fit3$draws("mu")
sigma_samples <- stan_fit3$draws("sigma")
precision_samples <- 1 / (sigma_samples**2)
```

```
## Make the plot
plot(x = mu_samples, y = precision_samples, main = 'mu Samples vs Precision Samples', xlab = 'mu Value'
```




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

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

With higher precision, the range of mu is smaller, ie the points with higher precision are closer together. The points with lower precision have a larger range, hence the diamond shaped plot. This makes sense because precision is $\frac{1}{\sigma^2}$, so lower variance means higher precision and that means the points will be closer together and near the mean. With high variance, precision is lower, and that means points will be scattered and range further 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)
```

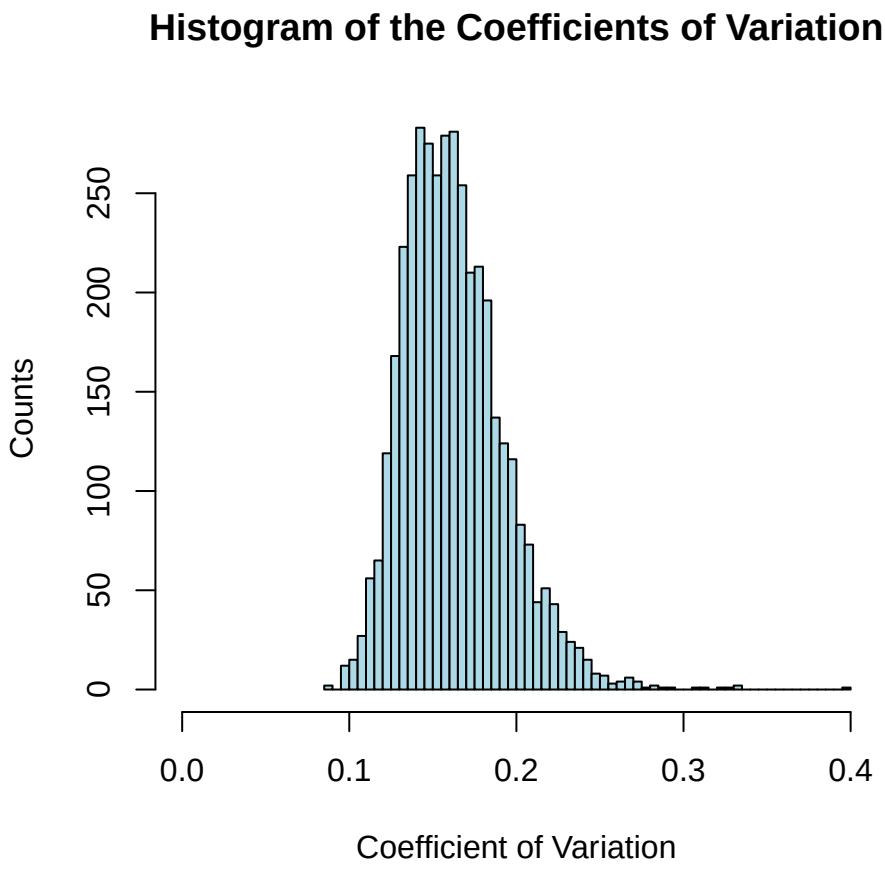
```
## [1] 0.7765
```

The posterior probability that μ is greater than 100 is 0.764

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.

```
# Calculating the coeffient of variation
coef_var <- sigma_samples / mu_samples
```

```
# Histogram
hist(coef_var, main = 'Histogram of the Coefficients of Variation', ylab = 'Counts', xlab = 'Coefficien
```




```
stan_fit3$summary()[2,2]  # Posterior Mean
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1  104.
```

```
# Posterior Credible Interval of Coefficient of Variation
c(quantile(coef_var, .025), quantile(coef_var, .975))
```

```
##      2.5%     97.5%
## 0.1140526 0.2304714
```

```
# Posterior Credible Interval of mu samples
c(quantile(mu_samples, .025), quantile(mu_samples, .975))
```

```
##      2.5%     97.5%
##  92.70668 115.58047
```

Not very clear if the question is asking for the posterior credible interval of the mu samples or the coefficient of variation, this was also asked on nectir and there was no response so I included both.

The 95% posterior credible interval for the coefficient of variation is [0.1129044, 0.2351031]

The posterior mean IQ score is 104.3191 and the lower and upper bounds of the 95% posterior credible interval are 92.8024 and 116.1633 respectively.