Maggie O’Shea
Bayesian Statistical Modeling and Computation
Professor Klaus Keller
Problem Set 2
January 24, 2025

1. Review an example script we discussed in class.

- Reviewed: coin-example.R

2. Review at least two other example scripts (for example from code replication repositories from papers in your application area)

- Other scripts reviewed:
    - Code from: Callahan, C. W. & Mankin, J. S. Globally unequal effect of extreme heat on economic growth. Science Advances 8, eadd3726 (2022).
        - https://github.com/ccallahan45/CallahanMankin_ENSOEconomics/tree/main
            - https://github.com/ccallahan45/CallahanMankin_ENSOEconomics/blob/main/Scripts/Plot_Teleconnection_Heterogeneity.ipynb
    - Code from: Gottlieb, A. R. & Mankin, J. S. Evidence of human influence on Northern Hemisphere snow loss. Nature 625, 293–300 (2024).
        - https://github.com/alex-gottlieb/swe_da/blob/main/scripts/fig4_sensitivity.py

3. Review the key sources already assigned as reading with a special focus on:

- Labs 0 to 3 in: Applegate, P. J., & Keller, K. (Eds.). (2016). Risk analysis in the Earth Sciences: A Lab manual. 2nd edition. Leanpub. Retrieved from https://leanpub.com/raes

4. Use a Monte Carlo simulation method to:

a. **Determine the mean and the 95 percentile from a known univariate normal distribution with a mean of zero and a standard deviation of one with your estimated uncertainties**

In order to determine the mean and 95th percentile from known univariate normal distribution, I randomly sampled using random sampling in programming software R (Version 2024.12.0+467) from a normal distribution with a mean of zero and standard deviation of one. A relevant assumption of this analysis is that R really does do random sampling with the code used. Figure 1 shows an example of the results from doing so with 1,000,000 samples and a random number generator based on seed 930. I selected 1,000,000 samples to start with a high number of samples in order to ensure there were enough samples to approach the known true mean (due to the law of large numbers and central limit theorem). However, I explored the other plausible choices in sample size below.

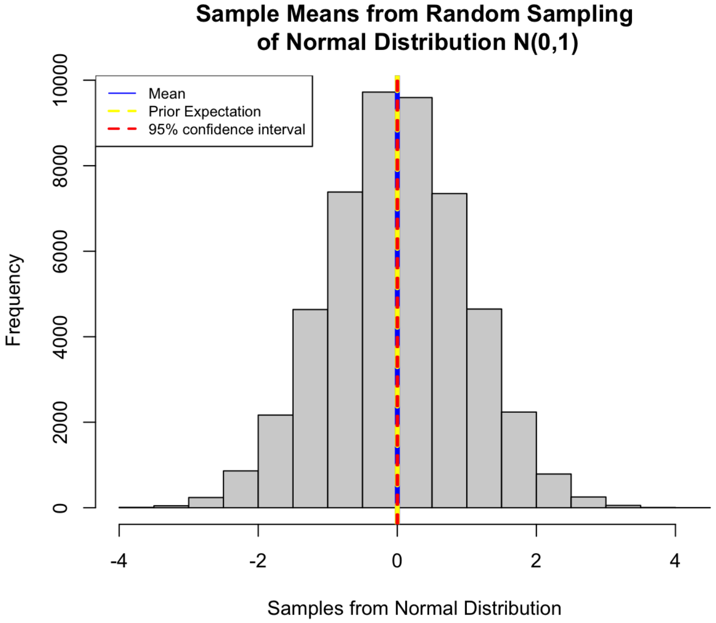

Figure 1: Sample means from 1,000,000 samples from univariate normal distribution with seed 930.

This estimation finds a mean of 0.0011 which is very slightly higher than the prior expectation of a mean of 0. The 95% confidence interval here is between -0.0076 to 0.0098, and thus contains the true mean (0). However, this estimation is not without uncertainties associated with it. In particular, there is uncertainty associated with the number of samples taken – ie with a different number of samples, would one get the same result? There is also uncertainty associated with the seed – ie with a different seed, would one get the same result? In order to investigate this, we can use different sample sizes as well as different seeds to characterize these uncertainties (Figure 2). Here we use 100 different seeds and sample sizes from 1,000 to 50,000, with 5,000 total different runs. This range choice was based on a tradeoff between trying as many sample sizes and seeds as possible against the computational time it takes to run each estimate.

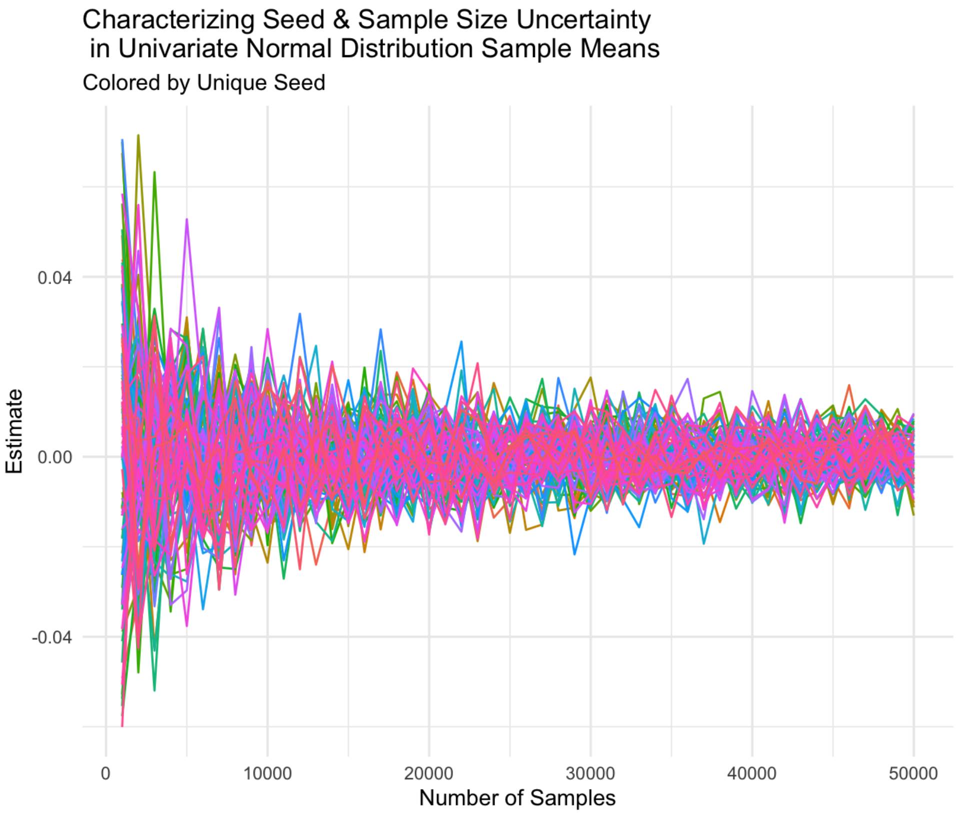

Figure 2: Sample Mean estimate by number of samples and lines colored by seed.

Figure 2 shows that depending on the number of samples, there can be variation in the estimated sample mean, as well as the range in results depending on the seed. Looking just at sample size, with the lowest number of samples, 1,000 samples, we see that the smallest sample mean was -0.0600 (seed 1025) and the largest sample mean was 0.0706 (seed 997). If we consider a threshold for convergence to be within 0.001 of the truth, in this case 0, the lowest sample size to be within 0.001 of the truth is for seed XX with XX samples. The seed with the largest sample size to reach within 0.001 of the true XX. highest sample size to first reach that is The standard deviation across all sample means was 0.0094, and the overall mean was -0.00001. All together this suggests that while there is uncertainty, the mean is relatively tight around the expected 0 mean, though this is less the case with fewer samples. The figure (Figure 2) shows that the values begin to converge more tightly around 0 as soon as 20,000 samples, though the variability around the mean due to sample size and seed continues to improve beyond that as well. This is better seen in Figure 3 where the confidence intervals and mean are calculated for each seed and then averaged across the 100 seeds.

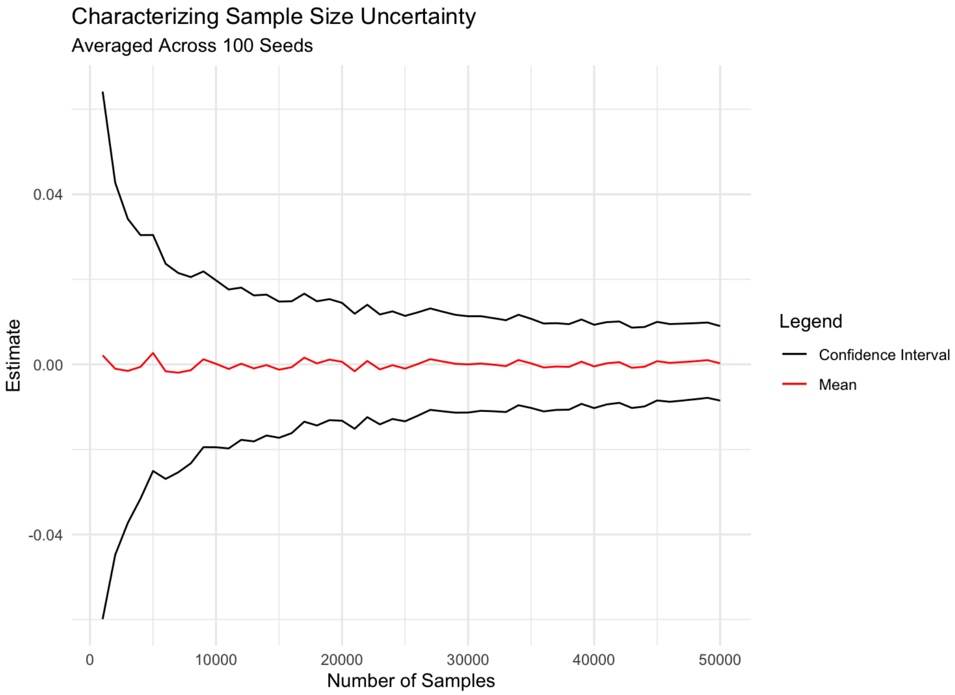

Figure 3: Characterizing Sample Size Uncertainty by averaging 95% confidence interval upper and lower bounds across 100 seeds and average mean across 100 seeds.

The convergence I am referring to a visual estimation of convergence, however, quantitatively it is also the case that by 35,000 samples the difference between the estimated mean and 0 is less than or equal to 0.0001. If using this as the threshold, then convergence occurs at 35,000 samples. Finally, this was made to be reproducible by including commented code as well as by specifying specific seeds even for the uncertainty estimations such that one could re-run and get the same results.

b. Determine the value of pi with your estimated uncertainties

In order to estimate the value of pi (π), we can take advantage of area equations that use pi, namely the area of a circle. We first imagine a circle inside a square in which the edges align such that the diameter of the circle is equal to the length of one side of the square. So the area of this square is D^2 where D is the diameter of the circle. Because the diameter is just $2r$, This can also be written as $(2r)^2$, which would simplify to $4r^2$. The area of the circle is πr^2. Taken together, the proportion of the area of the circle to the area of the square is $\frac{\pi r^2}{4r^2}$ and given that the squared radius values cancel, we can understand that the ratio of the area of the circle to the area of the square (where the diameter of the circle equals the length of one side of the square) is

$\frac{\pi}{4}$. Now, we can use this relationship to estimate a numeric value for the proportion of the circle area to square to then solve for π.

If the radius of the circle is 1, then we can imagine this circle and square centered around the origin of a graph, where the center of the circle and square is at point (0,0). I chose 1 for the radius of the circle for simplicity, but another value could be used here and it wouldn't compromise the approach. Using random sampling, we can determine the proportion of the area of the square that is covered by the circle by dividing the number of points that land within the circle by the number of total points sampled. Importantly, a relevant assumption of this analysis, similar to question A, is that R software really does do random sampling. With this proportion of circle:square, we can then set our estimated proportion to $\frac{\pi}{4}$ and solve for π.

In performing this analysis, the sources of uncertainty are from the seed used for the random sampling as well as the number of samples from the circle-square space used to estimate the proportion. The average value of π estimated across estimates from 200 different seeds and 10,00 different sample sizes was 3.1417. Figure 4 shows the different estimates of π found when using different seeds as well as different sample sizes which ranged from 100 to 10,000. Similar to question A, I selected this range and this choice was based on a tradeoff between trying out as many sample sizes and seeds as possible against the computational time it takes to run each estimate of pi. My original estimates were based on sample sizes from 100 to 5,000 and 100 unique seeds, however when taking the average pi estimate and confidence intervals around those sample sizes across seeds, there was still variation about the mean, suggesting perhaps that convergence hadn't yet happened at 5,000 with 100 seeds. So, I increased the maximum sample size to 10,000 as well as the seeds to 200 seeds.

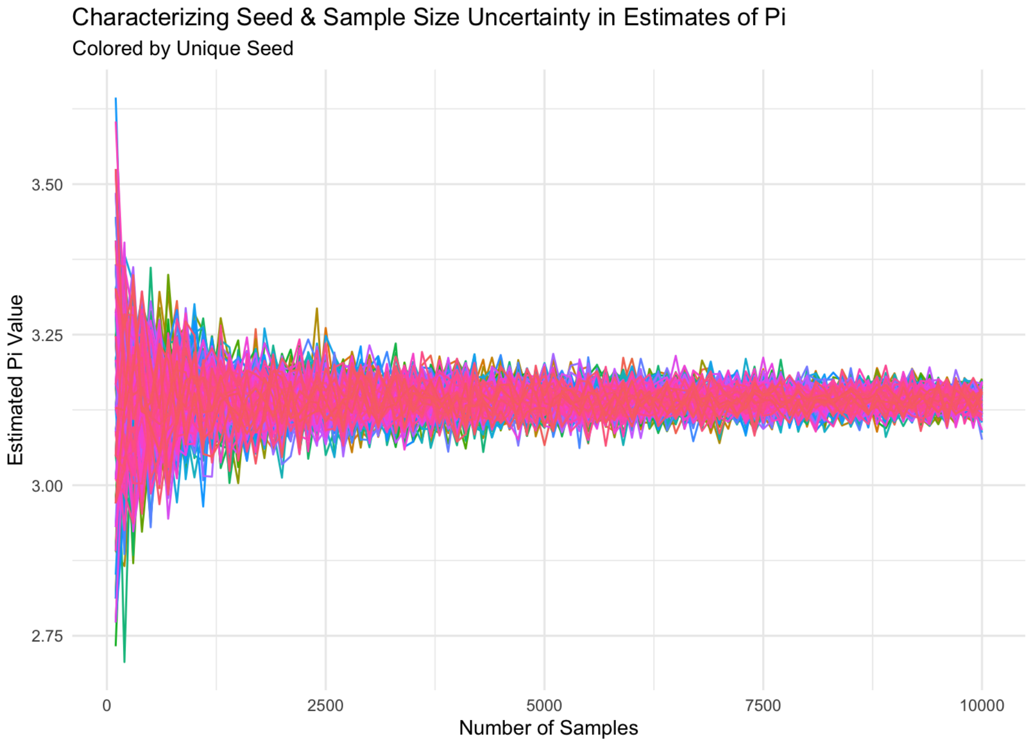

Figure 4: Estimates of Pi based on 200 different seeds and 1,000 different sample sizes for each seed (sample size ranged from 100, 200, 300 and so on up to 10,000).

This plot, figure 4, shows that the mean estimate for pi narrows around 3.14 as the sample size is increased. There is uncertainty about the mean associated with sample size and seed but it narrows closer to an average pi estimate of 3.14. By taking the average across all seeds at each sample size, we can visualize the sample size uncertainty without the different seeds shown.

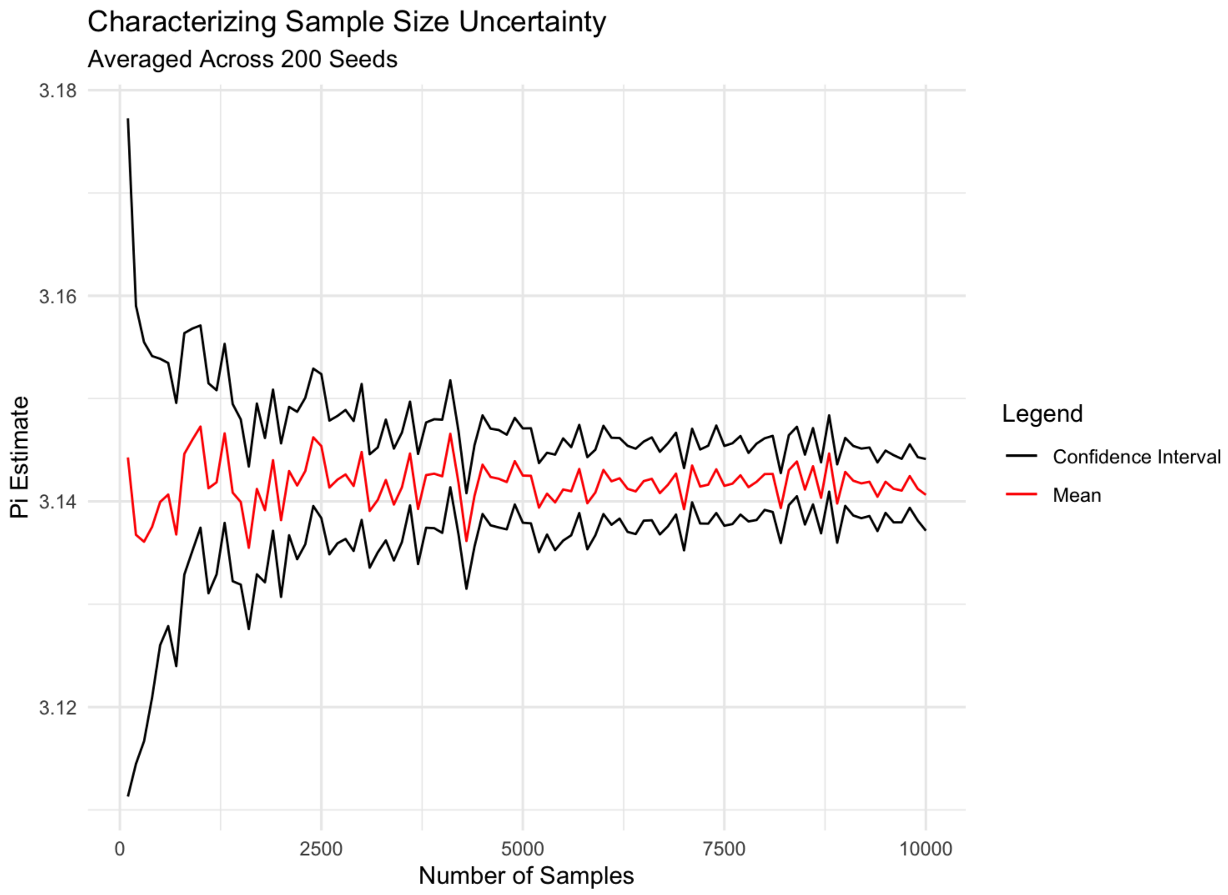

Figure 5: The average estimate of pi for each sample size across 200 seeds and the 95% confidence interval around this mean.

Looking across all seeds (Figure 4), the earliest convergence to 3.14 (within 0.001 of true pi) is at a sample size of just 400 (seed 939), and the latest convergence to within 0.001 of true pi is at a sample size of 9,900 (seed 1048). This shows that there is uncertainty even at the point of convergence based on the seed chosen (as well as the threshold chosen). With a looser threshold the point of convergence would be achieved with smaller sample sizes. Figure 5 shows that averaged across all seeds the mean pi converges to within 0.001 of true pi as soon as a sample size of 600, however it is not consistently below the threshold until a sample size of 9500. Finally, this code is included alongside question A's code and thus was made to be reproducible with similar commenting strategy, specifying specific even for the uncertainty estimations such that one could re-run and get the same results, and being uploaded to github.

Works Cited:

Applegate, P. J., & Keller, K. (Eds.). (2016). Risk analysis in the Earth Sciences: A Lab manual. 2nd edition. Leanpub. Retrieved from https://leanpub.com/raes

Callahan, C. W. & Mankin, J. S. Globally unequal effect of extreme heat on economic growth. Science Advances 8, eadd3726 (2022).

Gottlieb, A. R. & Mankin, J. S. Evidence of human influence on Northern Hemisphere snow loss. Nature 625, 293–300 (2024).

Appendix I:

Code written in software R (Version 2024.12.0+467).

- Downloaded from: https://www.r-project.org/

Saved to Github Repository:
https://github.com/maggieoshea/BayesianStatisticalModelingandComputation

Full R Script below.

```
############################################
##  file: MOShea_PS2.R
## Written on R Version 2024.12.0+467
######################################################
##  Maggie O'Shea
##  copyright by the author
##  distributed under the GNU general public license
##  https://www.gnu.org/licenses/gpl.html
##  no warranty (see license details at the link above)
######################################################
##   Course: Bayesian Statistical Modeling & Computation
##   Professor Klaus Keller
##   January 24, 2025
##   Problem Set #2
##################################################
# contact: margaret.oshea.gr@dartmouth.edu
##################################################
# sources:
# - Applegate, P. J., & Keller, K. (Eds.). (2016). Risk analysis in 
the Earth Sciences: A Lab manual. 2nd edition. Leanpub. Retrieved from 
https://leanpub.com/raes
# - Callahan, C. W. & Mankin, J. S. Globally unequal effect of extreme 
heat on economic growth. Science Advances 8, eadd3726 (2022).
# - Gottlieb, A. R. & Mankin, J. S. Evidence of human influence on 
Northern Hemisphere snow loss. Nature 625, 293–300 (2024).
# - coin.R file from 'Bayesian Stats' course with Dr. Klaus Keller
# - R help files accessed through R-studio for syntax
# - Class discussions in Bayesian Statistical Modeling & Computation, 
Dartmouth College, Winter 2025
# - Online R coding help, namely:
#   - For loops: https://www.w3schools.com/r/r_for_loop.asp
#   - Binary Variable: https://stackoverflow.com/questions/19970009/
create-binary-column-0-1-based-on-condition-in-another-column
#   - Slice_min: https://stackoverflow.com/questions/24070714/extract-
row-corresponding-to-minimum-value-of-a-variable-by-group
# - Conversation with Alexis Hudes (specifically for B) and Anna 
Valentine (about convergence)
####################################################

## Packages ##
# If packages ggplot2 and dplyr not already downloaded, un-tag (remove 
the #) to download packages before running the rest of the script.
#install.packages("ggplot2")
#install.packages("dplyr")
library(ggplot2)
library(dplyr)

### Instructions: ###
## Question 4A. Use Monte Carlo method to determine the mean and 95 
Percentile from known univariate normal distribution: Mean of zero and
```

SD of one with one seed and one sample size:

```
# define a seed for reproducibility
set.seed(930)

# Number of trials
n_trials <- 10^5

samples <- rnorm(n_trials, 0, 1)

# Find the mean #
mean = mean(samples)
print(mean)

# 95% confidence interval: #
standard_deviation = sd(samples)
sqrt_n = sqrt(n_trials)
CI_95_upper = mean+(1.96*(standard_deviation/sqrt_n))
CI_95_lower = mean-(1.96*(standard_deviation/sqrt_n))
print(paste("95% Confidence Interval Upper Bound: ", CI_95_upper))
print(paste("95% Confidence Interval Lower Bound: ", CI_95_lower))

# Plot Results from one Sample #
hist(samples, main = "", xlab = "Samples from Normal Distribution")
abline(v=mean,col="blue",lty=1,lwd=4)
abline(v=0,col="yellow",lty=2,lwd=4)
abline(v=CI_95_lower, col="red", lty=2, lwd=2)
abline(v=CI_95_upper, col="red", lty=2, lwd=2)
legend("topleft", c("Mean","Prior Expectation",
                    "95% confidence interval"),
       lwd=c(1,2,2), lty=c(1,2,2), col=c("blue","yellow","red"),
cex=0.75)
title(main="Sample Means from Random Sampling\n of Normal Distribution
N(0,1)")

## Characterize Uncertainty from Sample Size and Seed ##

# Vary seed and sample size
seeds <- seq.int(930, 1030, by = 1) # 100 seeds
sample_sizes <- seq.int(1000, 50000, by = 1000) # 500 sample sizes

# Prepare dataframe to save results from each seed and sample size
combination
mean_df <- data.frame(
  seed = integer(),
  sample_size = integer(),
  mu = numeric(),
  CI_upper = numeric(),
  CI_lower = numeric()
)
```

```
# for each seed
for (seed in seeds) {
  set.seed(seed)

  # for each sample size
  for (size in sample_sizes){
    samples <- rnorm(size, 0, 1)

    # calculate the mean
    mean = mean(samples)

    # calculate CI
    standard_deviation = sd(samples)
    sqrt_n = sqrt(size)
    CI_95_upper = mean+(1.96*(standard_deviation/sqrt_n))
    CI_95_lower = mean-(1.96*(standard_deviation/sqrt_n))

    # append this data to the dataframe as a new row
    mean_df <- rbind(mean_df, data.frame(seed = seed, sample_size =
size, mu = mean, CI_upper = CI_95_upper,  CI_lower = CI_95_lower))


  }
}

# Plot with ggplot2 the results from 500 sample sizes and 100 seeds
ggplot(data = mean_df, aes(x = sample_size, y =mu, color =
factor(seed), group = seed)) +
  geom_line() +
  labs(title = "Characterizing Seed & Sample Size Uncertainty\n in
Univariate Normal Distribution Sample Means",
       subtitle = "Colored by Unique Seed",
       x = "Number of Samples",
       y = "Estimate",
       color = "Seed") +
  theme_minimal() +
  theme(legend.position = "none")

## Creating a "mean" dataframe which takes the average CI and mean
across all seeds
CI_mean <- mean_df %>%
  group_by(sample_size) %>%
  summarise(avg_upperCI = mean(CI_upper),
            avg_lowerCI = mean(CI_lower),
            avg_mu = mean(mu))

CI_mean$differencefromtruth = (0-CI_mean$avg_mu)

colors <- c("Mean" = "red", "Confidence Interval" = "black")
```

```
ggplot(data = CI_mean, aes(x = sample_size)) +
  geom_line(data = CI_mean, aes(y = avg_upperCI))+
  geom_line(data = CI_mean, aes(y = avg_lowerCI, color = 'Confidence
Interval'))+
  geom_line(data = CI_mean, aes(y = avg_mu, color = 'Mean'))+

  labs(title = "Characterizing Sample Size Uncertainty",
       subtitle = "Averaged Across 100 Seeds",
       x = "Number of Samples",
       y = "Estimate",
       color = "Legend") +
  theme_minimal() +
  theme(legend.position = "right")   +
  scale_color_manual(values = colors)

### Instructions: ###
## Quesion 4B. Determine the value of pi with your estimated
uncertainties ##
# Sources of uncertainty to characterize: sample size and seeds

# Vary number of samples and seeds
seeds <- seq.int(930, 1130, by = 1) # 200 seeds
sample_sizes <- seq.int(100, 10000, by = 100) # number of trials in
each seed sample

# Prep dataframe to hold results with different seeds and sample sizes
results_df <- data.frame(
  seed = integer(),
  sample_size = integer(),
  pi_estimate = numeric()
)

# for each seed
for (seed in seeds) {
  set.seed(seed)
  # for each sample size
  for (sample in sample_sizes){
    # sample size is the number of trials in each seed run
    # The number of trials is the number of samples in the circle/
square space
    sample_size=sample
    n_trials <- seq.int(0, sample_size, by = 1)

    # prep circle and square variables to be able to add up the number
of circle points vs square points
    circle <- 0
    square <- 0
    for (trial in n_trials) {
      # include print statement if interested in tracking which trial
run
      #print(paste("running trial #", trial))
      n = 1 # 1 xy combo per trial

      # random number in uniform distribution from 1 to -1
      x_point <- runif(n, -1, 1)
      y_point <- runif(n, -1,1)

      # calculate the distance between random xy point and origin
      distance_from_origin = sqrt(x_point^2 + y_point^2)

      # if distance is less than radius 1, then it's in the circle,
otherwise in square
      if (distance_from_origin <= 1){
        circle = circle +1}
      else
        square = square +1 }


    # proportion of circle in this space.
    # The denominator is square + circle because anywhere in circle is
also in square
    circle_prop = circle/(square+circle)
    pi_value = circle_prop*4

    # append results from this run to the dataframe for results
    results_df <- rbind(results_df, data.frame(seed = seed,
sample_size = sample_size, pi_estimate = pi_value))


  }
  }


# Estimating Convergence...
# Find the smallest sample size at which point the different between
the truth is within 0.001
earliest_meeting_threshold <- results_df %>%
  group_by(seed) %>%  # Group by the seed column
  filter(pi_estimate >= pi - 0.001 & pi_estimate <= pi + 0.001) %>%
  slice_min(order_by = sample_size) # filter for only columns that
meet threshold


# plot results from each seed and sample size using ggplot2
ggplot(data = results_df, aes(x = sample_size, y = pi_estimate, color
= factor(seed), group = seed)) +
  geom_line() +
  labs(title = "Characterizing Seed & Sample Size Uncertainty in
Estimates of Pi",
       subtitle = "Colored by Unique Seed",
       x = "Number of Samples",
       y = "Estimated Pi Value",
       color = "Seed") +  # Label for the color legend
  theme_minimal() +
  theme(legend.position = "none")

# calculate mean pi estimates and CI across all seeds, grouped by
sample sizes
mean_pi_df <- results_df %>%
  group_by(sample_size) %>%
  summarise(sd_pi = sd(pi_estimate),
            mean_pi = mean(pi_estimate),
            # sqrt(100) is the sqrt of n, the number of seeds per
sample size
            avg_upperCI = mean_pi+(1.96*sd_pi/(sqrt(100))),
            avg_lowerCI = mean_pi-(1.96*sd_pi/(sqrt(100))))

# Find where the pi estimate is within a threshold of true pi (0.001)
mean_pi_df$difference_from_truepi = abs(pi-mean_pi_df$mean_pi)
mean_pi_df$meetsthreshold <-
ifelse(mean_pi_df$difference_from_truepi<=0.001, 1, 0)


# Plot results from average mean and CI by sample size
colors <- c("Mean" = "red", "Confidence Interval" = "black")

ggplot(data = mean_pi_df, aes(x = sample_size)) +
  geom_line(data = mean_pi_df, aes(y = avg_upperCI))+
  geom_line(data = mean_pi_df, aes(y = avg_lowerCI, color =
'Confidence Interval'))+
  geom_line(data = mean_pi_df, aes(y = mean_pi, color = 'Mean'))+
  # Add line for true pi
  #geom_hline(yintercept= pi, color = "blue", size=0.5) +

  labs(title = "Characterizing Sample Size Uncertainty",
       subtitle = "Averaged Across 200 Seeds",
       x = "Number of Samples",
       y = "Pi Estimate",
       color = "Legend") +
  theme_minimal() +
  theme(legend.position = "right")   +
  scale_color_manual(values = colors)
```