Problem Set 4

Michael Ghattas

Notes

Other students who I worked with on this assignment (if any): None.

(1 point)

Introduction

These questions were rendered in R markdown through RStudio (https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf, http://rmarkdown.rstudio.com).

Please generate your solutions in R markdown and upload a knitted pdf document to Gradescope. Please put your name in the "author" section in the header.

Background

Suppose $(S, M, P)_{\boldsymbol{\theta}}$ is a parametrized family of distributions. The parameter $\boldsymbol{\theta}$ may be vector-valued or one dimensional. Under fairly general circumstances, the maximum likelihood parameter estimate $\hat{\boldsymbol{\theta}}$ of the parameter $\boldsymbol{\theta}$ based on a sample $\{X_1, X_2, \ldots X_n\}$ is *consistent*, also called *asymptotically consistent*. Informally, this means that as larger and larger samples are used to estimate the parameter, the estimate gets closer and closer to the true value.

Some parameter estimates are *unbiased*. Informally, this means that if the estimate is applied to M samples of size n to get a collection of estimates $\left\{\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \ldots \hat{\boldsymbol{\theta}}_M\right\}$, the mean of the estimates, $\frac{1}{M}\sum_{i=1}^{M}\hat{\boldsymbol{\theta}}_i$ will get closer and closer to $\hat{\boldsymbol{\theta}}$ as M gets larger and larger.

In this assignment you will perform numerical experiments on samples from a $Normal(\mu, \sigma^2)$ distribution to see whether the maximum likelihood estimates for μ and σ^2 appear to be consistent and unbiased.

Question 1

Context

The purpose of this question is to perform numerical experiments to gain insight into the whether of maximum likelihood estimates of μ and σ^2 are consistent for samples from $Normal(0, 1)$.

The code provided generates $N = 500,000$ samples $\{x_1, x_2, \ldots x_N\}$ from the standard Normal distribution, $Normal(0, 1)$. For each value n in $\{1000, 2000, 3000, \ldots N\}$, the maximum likelihood estimates of μ and σ^2 are computed for the initial portion $\{x_1, x_2, \ldots x_n\}$ of the sample $\{x_1, x_2, \ldots x_N\}$. These values are stored in order of n in the data frame "dat.consist" with the variable names "mu.hat" and "sigma.sq.hat" respectively. A column of the corresponding values of n is added under the variable name "n". Below you will use this data frame to examine whether these samples provide numerical evidence that the maximum likelihood estimates of μ and σ^2 are consistent. Plotting using "geom_line" may be helpful.

```
set.seed(123456)
N<-500000
samp<-rnorm(N)
# function to compute the maximum likelihood estimate of mu and the sigma-
squared based on the first n values in a vector "samp" of samples from a
Normal distribution:
theta.est<-function(n,s=samp){
  m<-mean(s[1:n])
  s2<-sum((s[1:n]-m)^2)/n #
  return(c(m,s2))
}
dat.consist<-t(sapply(seq(1000,N,by=1000),theta.est))
dat.consist<-data.frame(dat.consist)
dat.consist$n<-seq(1000,N,by=1000)
names(dat.consist)<-c("mu.hat","sigma.sq.hat","n")
mean(dat.consist$mu.hat)
```

```
## [1] 0.001805303
```

Question 1.1

(2 points)

What is the true value of the parameter μ for these data? Please give a numeric value.

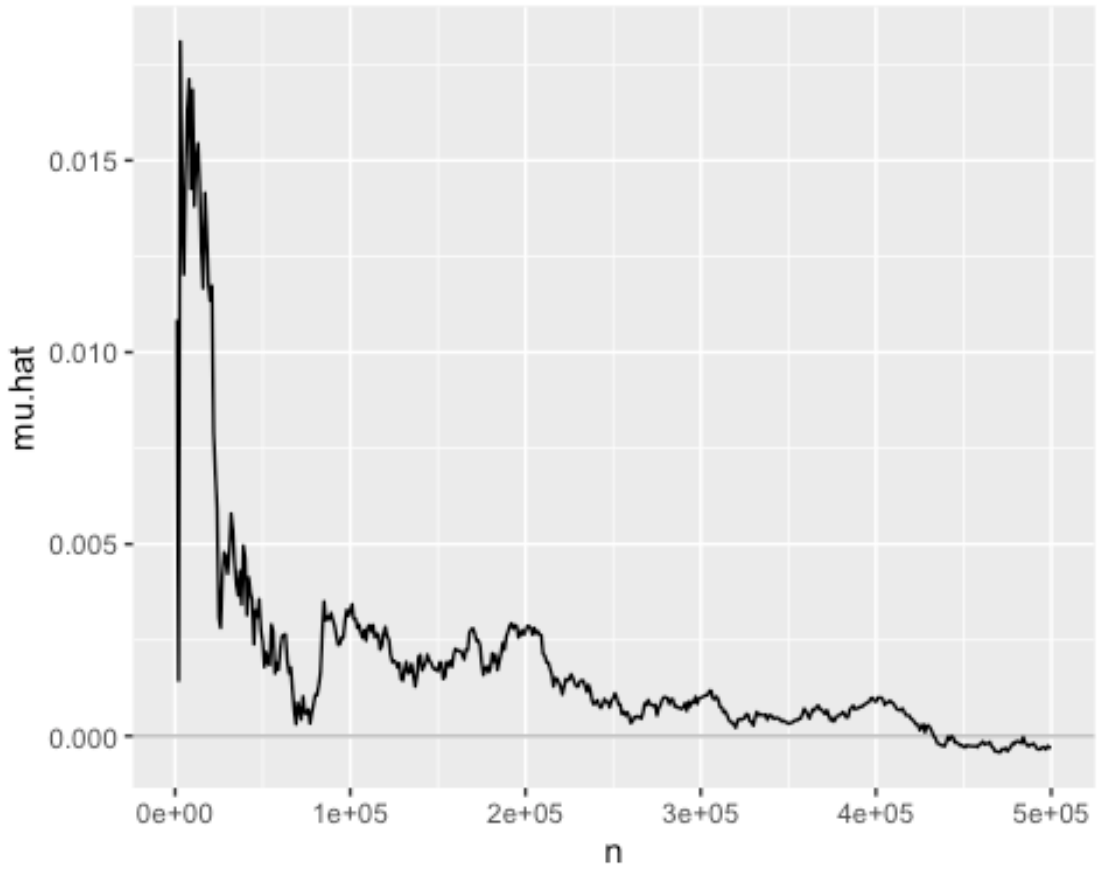
Your answer here: The true value of the parameter μ for these data is indeed 0 because we are dealing with a standard normal distribution N(0,1), where the mean is defined as 0. The mean of the sample estimates is close to 0 (0.001), which is expected due to the nature of sampling variability and the large sample size used in the estimation.

Question 1.2

(2 points)

Do the estimates of "mu.hat" of μ in "dat.consist" appear to approach the true value as the sample size "n" increases?

```
ggplot(dat.consist,aes(x=n,y=mu.hat))+
  geom_hline(yintercept = 0,color="gray")+
  geom_line()
```




```
dat.consist$mu.hat[c(1,5,10,50,100,500)]

## [1]  0.0108535170  0.0120273564  0.0168396665  0.0024677304  0.0032766567
## [6] -0.0002950421
```

Your answer here: Yes, the estimates of "mu.hat" in "dat.consist" do appear to approach the true value as the sample size increases. The plot shows a convergence of the estimates towards 0 as well.

Question 1.3

(2 points)

Does this numerical experiment suggest that the maximum likelihood estimate of μ is consistent?

Your answer here: Yes, this numerical experiment suggests that the maximum likelihood estimate of μ is consistent. The consistency of an estimator means that as the sample size increases, the estimates converge to the true parameter value. The plot shows that the estimates approach the true value of 0 as increases. Additionally, the values demonstrate that the estimates get closer to 0 with larger sample sizes, confirming that the maximum likelihood estimate of μ is consistent.

Question 1.4

(2 points)

What is the true value of the parameter σ^2 for these data? Please give a numeric value.

Your answer here: The true value of the parameter σ^2 for a standard normal distribution N(0,1) is $\sigma^2 = 1$.

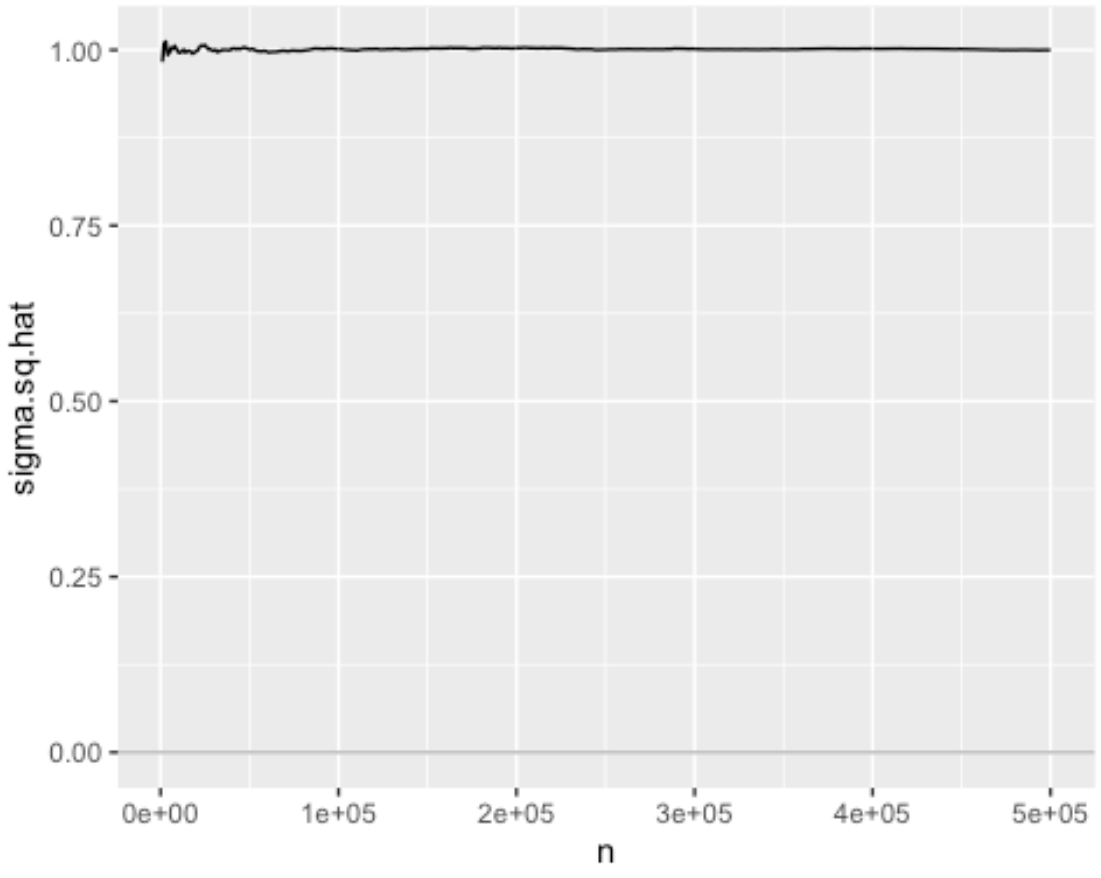
Question 1.5

(2 points)

Do the estimates of "sigma.sq.hat" of σ^2 in "dat.consist" appear to approach the true value as the sample size "n" increases? If you are unsure, you can calculate the estimate for some very large samples.

Your answer and code here: Yes, the estimates of "sigma.sq.hat" of σ^2 in "dat.consist" do appear to approach the true value as the sample size increases. The plot shows a convergence of the estimates towards 1 demonstrate this trend as well.

```
ggplot(dat.consist,aes(x=n,y=sigma.sq.hat))+
  geom_hline(yintercept = 0,color="gray")+
  geom_line()
```




```
dat.consist$sigma.sq.hat[c(1,5,10,50,100,500)]
```

```
## [1] 0.9828973 0.9961045 0.9967854 1.0002562 1.0015304 0.9999290
```

Question 1.6

(2 points)

Does this numerical experiment suggest that the maximum likelihood estimate of σ^2 is consistent?

Your answer here: Yes, this numerical experiment suggests that the maximum likelihood estimate of σ^2 is consistent. Consistency of an estimator means that as the sample size increases, the estimates converge to the true parameter value. The plot of shows that the estimates approach the true value of 1 as n increases.

Question 2

The purpose of this question is to perform numerical experiments to gain insight into whether the maximum likelihood estimates of μ and σ^2 are unbiased for samples of size 5 from $Normal(0, 1)$

Question 2.1

(1 point)

Create a $10,000 \times 5$ matrix of samples of size 5 from the standard Normal distribution.

Your code here:

```
set.seed(45678)
mat <- matrix(rnorm(10000 * 5),ncol = 5)
```

Question 2.2

(2 points)

Please use `apply` to calculate the maximum likelihood estimates $\hat{\mu}$ and $\hat{\sigma}^2$ of μ and σ^2 for each sample.

Your code here:

```
# Function to calculate MLE for mu and sigma^2
mle <- function(sample) {
  mu_hat <- mean(sample)
  sigma_sq_hat <- sum((sample - mu_hat)^2) / length(sample)
  return(c(mu_hat, sigma_sq_hat))
}

# Apply the MLE function to each row of the matrix
mle_estimates <- apply(mat, 1, mle)

# Convert the result to a data frame for easier manipulation
mle_estimates <- t(mle_estimates)
mle_estimates <- data.frame(mle_estimates)
names(mle_estimates) <- c("mu_hat", "sigma_sq_hat")
```

Question 2.3

(2 points)

Compute the mean of the $\hat{\mu}$s and the mean of the $\hat{\sigma}^2$s.

Your code here:

```
# Compute the mean of the estimates
mean_mu_hat <- mean(mle_estimates$mu_hat)
mean_sigma_sq_hat <- mean(mle_estimates$sigma_sq_hat)

# Display the results
mean_mu_hat
```

```
## [1] 0.008816334
```

```
mean_sigma_sq_hat
```

```
## [1] 0.8049875
```

Question 2.4

(3 points)

Does the maximum likelihood estimate of μ seem to be unbiased? (You may repeat the experiment with other seeds to help answer this question.)

Your answer and code here: The results suggest that the maximum likelihood estimate of $\mu$is unbiased. The mean of the values across different seeds is consistently close to the true value of $\mu = 0$, supporting the conclusion that the MLE for μ is unbiased.

```
# Function to calculate MLE for mu and sigma^2
mle <- function(sample) {
  mu_hat <- mean(sample)
  sigma_sq_hat <- sum((sample - mu_hat)^2) / length(sample)
  return(c(mu_hat, sigma_sq_hat))
}

# Function to perform the experiment and calculate mean estimates
perform_experiment <- function(seed) {
  set.seed(seed)
```

```r
  mat <- matrix(rnorm(10000 * 5), ncol = 5)
  mle_estimates <- apply(mat, 1, mle)
  mle_estimates <- t(mle_estimates)
  mle_estimates <- data.frame(mle_estimates)
  names(mle_estimates) <- c("mu_hat", "sigma_sq_hat")

  mean_mu_hat <- mean(mle_estimates$mu_hat)
  mean_sigma_sq_hat <- mean(mle_estimates$sigma_sq_hat)

  return(c(mean_mu_hat, mean_sigma_sq_hat))
}

# Perform the experiment with different seeds
results <- t(sapply(c(12345, 67890, 98765, 54321), perform_experiment))
colnames(results) <- c("mean_mu_hat", "mean_sigma_sq_hat")

results
```

```
##        mean_mu_hat mean_sigma_sq_hat
## [1,]  0.0037738278         0.7930296
## [2,]  0.0029808920         0.7918856
## [3,] -0.0011327822         0.7908439
## [4,] -0.0001739905         0.8091933
```

Question 2.5

(3 points)

Does the maximum likelihood estimate of σ^2 seem to be unbiased? (You may repeat the experiment with other seeds to help answer this question. Try comparing with the adjusted estimates produced by dividing the sum of the squared differences by 4 instead of 5.)

Your answer and code here: Based on the numerical experiment with different seeds, the maximum likelihood estimate of σ^2 does not seem to be unbiased when using the MLE formula (dividing by n). The adjusted estimates, produced by dividing the sum of squared differences by $n - 1$ (which is 4 in this case), provide unbiased estimates as they are consistently closer to the true value of $\sigma^2 = 1$.

```r
rm(mat) # remove matrix variable from environment

# Function to calculate MLE for mu and sigma^2
mle <- function(sample) {
  mu_hat <- mean(sample)
  sigma_sq_hat <- sum((sample - mu_hat)^2) / length(sample)
  return(c(mu_hat, sigma_sq_hat))
}

# Function to calculate adjusted estimates for sigma^2
adjusted_sigma_sq <- function(sample) {
  mu_hat <- mean(sample)
  sigma_sq_adjusted <- sum((sample - mu_hat)^2) / (length(sample) - 1)
  return(sigma_sq_adjusted)
}

# Function to perform the experiment and calculate mean estimates
perform_experiment <- function(seed) {
  set.seed(seed)
  mat <- matrix(rnorm(10000 * 5), ncol = 5)
  mle_estimates <- apply(mat, 1, mle)
  adjusted_estimates <- apply(mat, 1, adjusted_sigma_sq)

  mle_estimates <- t(mle_estimates)
  mle_estimates <- data.frame(mle_estimates)
  names(mle_estimates) <- c("mu_hat", "sigma_sq_hat")

  mean_mu_hat <- mean(mle_estimates$mu_hat)
  mean_sigma_sq_hat <- mean(mle_estimates$sigma_sq_hat)
  mean_adjusted_sigma_sq <- mean(adjusted_estimates)

  return(c(mean_mu_hat, mean_sigma_sq_hat, mean_adjusted_sigma_sq))
}

# Perform the experiment with different seeds
results <- t(sapply(c(12345, 67890, 98765, 54321), perform_experiment))
```

```r
colnames(results) <- c("mean_mu_hat", "mean_sigma_sq_hat",
"mean_adjusted_sigma_sq")

results
```

```
##        mean_mu_hat mean_sigma_sq_hat mean_adjusted_sigma_sq
## [1,]  0.0037738278         0.7930296              0.9912870
## [2,]  0.0029808920         0.7918856              0.9898570
## [3,] -0.0011327822         0.7908439              0.9885548
## [4,] -0.0001739905         0.8091933              1.0114916
```

Question 3

Context

The uniform distributions are a two parameter family of continuous distributions, $Uniform(a, b)$ with $a, b \in \mathbb{R}$ and $a < b$. Given (a, b), the sample space is $[a, b]$ and the probability density function is $f(x) = \frac{1}{b-a}$.

Question 3.1

(5 points)

Please compute the mean of $Uniform(a, b)$.

Your answer here:

Question 3.2

(5 points)

Please compute the variance of $Uniform(a, b)$. The identity

$$b^n - a^n = (b-a)\sum_{k=0}^{n-1} b^{n-1-k}a^k$$

may be useful in simplifying the formula.

Your answer here: To find the mean of the $Uniform(a, b)$ distribution, we start with the probability density function (PDF):

$$f(x) = \frac{1}{b-a}, \quad \text{for } a \le x \le b\,.$$

The mean (expected value) of X is given by:

$$\mathbb{E}[X] = \int_a^b x f(x)\,dx\,.$$

Substituting the PDF $f(x) = \frac{1}{b-a}$:

$$\mathbb{E}[X] = \int_a^b x \cdot \frac{1}{b-a}\,dx\,.$$

Let's compute this integral:

$$\begin{aligned}
\mathbb{E}[X] &= \tfrac{1}{b-a}\textstyle\int_a^b x\,dx \\
&= \tfrac{1}{b-a}\left[\tfrac{x^2}{2}\right]_a^b \\
&= \tfrac{1}{b-a}\left(\tfrac{b^2}{2} - \tfrac{a^2}{2}\right) \\
&= \tfrac{1}{b-a} \cdot \tfrac{b^2-a^2}{2} \\
&= \tfrac{1}{b-a} \cdot \tfrac{(b-a)(b+a)}{2} \\
&= \tfrac{b+a}{2}\,.
\end{aligned}$$

Thus, the mean of the $Uniform(a, b)$ distribution is:

$$\mathbb{E}[X] = \frac{a+b}{2}\,.$$

Question 4

Context

The data sets in these questions were downloaded 6/13/2022 from https://ourworldindata.org/

The code chunks below read in a data frame of world populations and a data frame of world population densities.

```
dat.pop<-read.csv("population-since-1800.csv",stringsAsFactors = FALSE)
names(dat.pop)[4]<-"population"
dat.den<-
  read.csv("population-density.csv")
names(dat.den)[4]<-"density"
```

Question 4.1

(4 points)

Write code to restrict both data frames to cases in which the value of "Year" is 2020 and the value of "Code" is not the empty string, "", and is not the value for the whole world," OWID_WRL". Please display the number of rows in the resulting data frames using the function `nrow`.

Your code here:

```
# Filter the data frames
dat.pop_2020 <- subset(dat.pop, Year == 2020 & Code != "" & Code !=
"OWID_WRL")
dat.den_2020 <- subset(dat.den, Year == 2020 & Code != "" & Code !=
"OWID_WRL")

# Display the number of rows in the resulting data frames
nrow(dat.pop_2020)

## [1] 234

nrow(dat.den_2020)

## [1] 216
```

The following code merges the data sets, restricting to values of "Code" occurring in both data sets.

```
dat.both<-inner_join(dat.den,dat.pop,by="Code")

## Warning in inner_join(dat.den, dat.pop, by = "Code"): Detected an
unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
```

```
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.

# check
mean(dat.both$Entity.x==dat.both$Entity.y)

## [1] 0.9738883
```

Question 4.2

(4 points)

Write code to find the four indices in "dat.both" at which the population takes on its minimum or maximum value and at which the density takes on its minimum or maximum value. Store the resulting indices in a vector named "inds". Use of the `which` function can simplify this effort. The functions `which.min` and `which.max` may also be used. Please display the "Entity.x" values of the identified rows.

Your code here:

```
# Find the indices of the minimum and maximum values of population and
density
min_pop_index <- which.min(dat.both$population)
max_pop_index <- which.max(dat.both$population)
min_den_index <- which.min(dat.both$density)
max_den_index <- which.max(dat.both$density)

# Store the indices in a vector named "inds"
inds <- c(min_pop_index, max_pop_index, min_den_index, max_den_index)

# Display the "Entity.x" values of the identified rows
dat.both$Entity.x[inds]

## [1] "Turks and Caicos Islands" "World"
## [3] "Angola"                   "Monaco"
```

Question 4.3

(4 points)

Use “transmute” from dplyr to modify “dat.both” to be a data frame based on “dat.both”, but with the value of “Entity.x” in a variable labeled “entity”, the log of “density” in a variable labeled “den.log”, and the log of “Population” in a variable labeled “pop.log” and no other variables. Please display first 5 rows of the new version of “dat.both”.

Your code here:

```
# Assuming the data is already loaded and processed as before

# Use transmute to create a new data frame with the specified variables
dat.both_modified <- dat.both %>%
  transmute(entity = Entity.x, den.log = log(density), pop.log =
log(population))

# Display the first 5 rows of the new version of dat.both
head(dat.both_modified, 5)

##        entity   den.log  pop.log
## 1 Afghanistan -3.772261 15.00335
## 2 Afghanistan -3.772261 15.00335
## 3 Afghanistan -3.772261 15.00335
## 4 Afghanistan -3.772261 15.00335
## 5 Afghanistan -3.772261 15.00335
```

Create and display a data frame “dat.text” from dat.both that includes only the rows containing the extremes identified in question 4.3.

Your code here:

```
# Create and display a data frame "dat.text" from dat.both that includes only
the rows containing the extremes
dat.text <- dat.both_modified[inds, ]

# Display the resulting dat.text data frame
dat.text

##                            entity   den.log   pop.log
## 14152954 Turks and Caicos Islands -6.907755  5.159055
## 15000327                   World -3.381395 22.786955
```

```
## 316936                      Angola      -Inf 14.264691
## 8664025                     Monaco 10.365753  8.960468
```

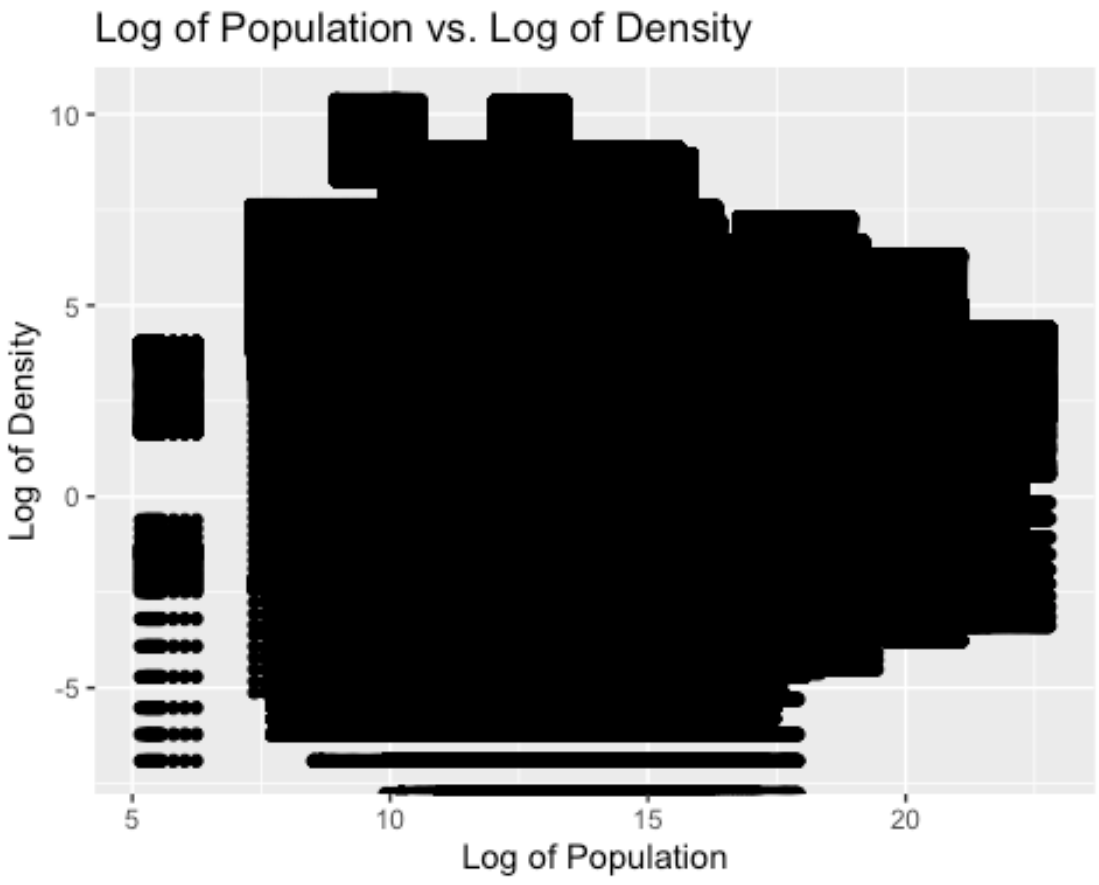
Question 4.4

(4 points)

Use “ggplot” to create a point plot of the log of population (on the x-axis) versus the log of density. Store the plot in the variable g. Display the plot.

Your code here:

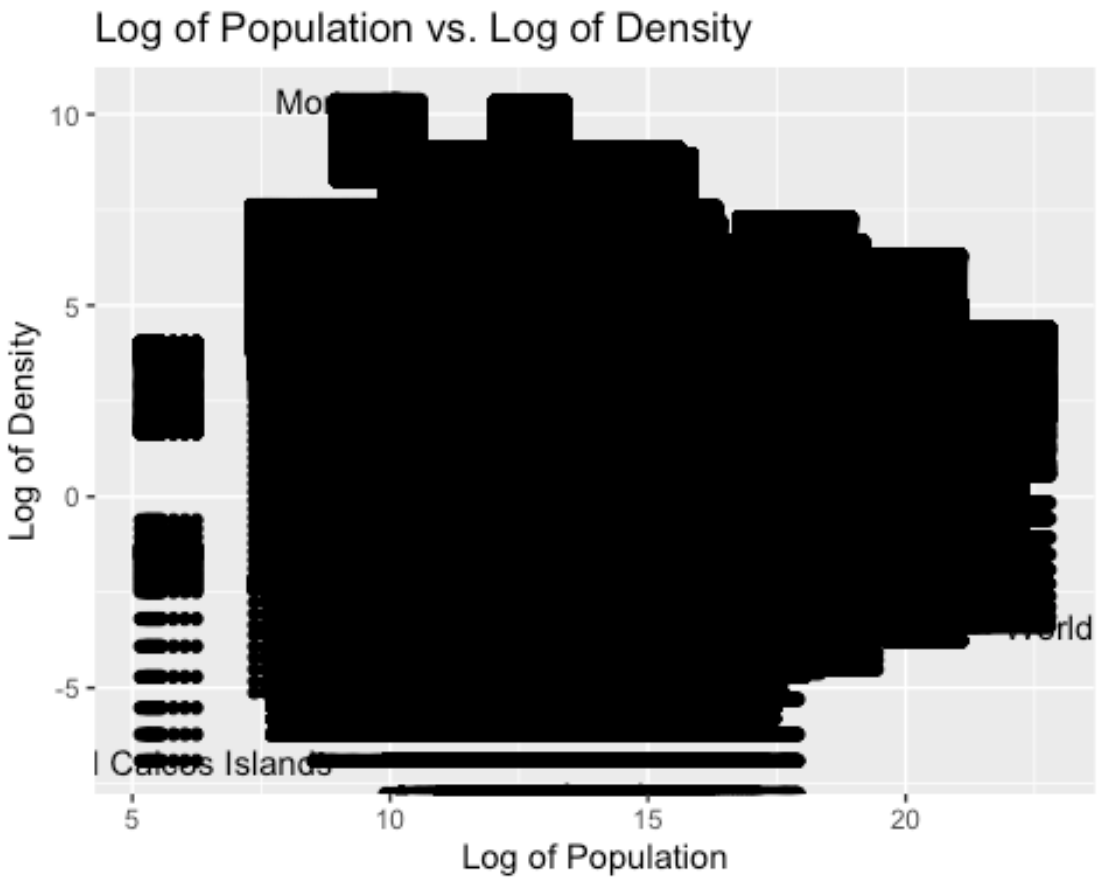
```
library(ggplot2)

# Create the ggplot and save it to a file
g <- ggplot(dat.both_modified, aes(x = pop.log, y = den.log)) + geom_point()
+
  labs(x = "Log of Population", y = "Log of Density", title = "Log of
Population vs. Log of Density")

# Display the plot
print(g)
```



The following should give the previous plot with the names of the entities having extreme population or extreme density, assuming that the result of the "transmute" call was stored back in "dat.both".

Uncomment and run code here:

```
# Please uncomment and run:
g = g + geom_text(data = dat.text, aes(x = pop.log, y = den.log, label =
entity))
g
```


Log of Population vs. Log of Density
Log of Density
Log of Population
10
5
0
-5
5
10
15
20