Homework 4

Shivam Bajaj

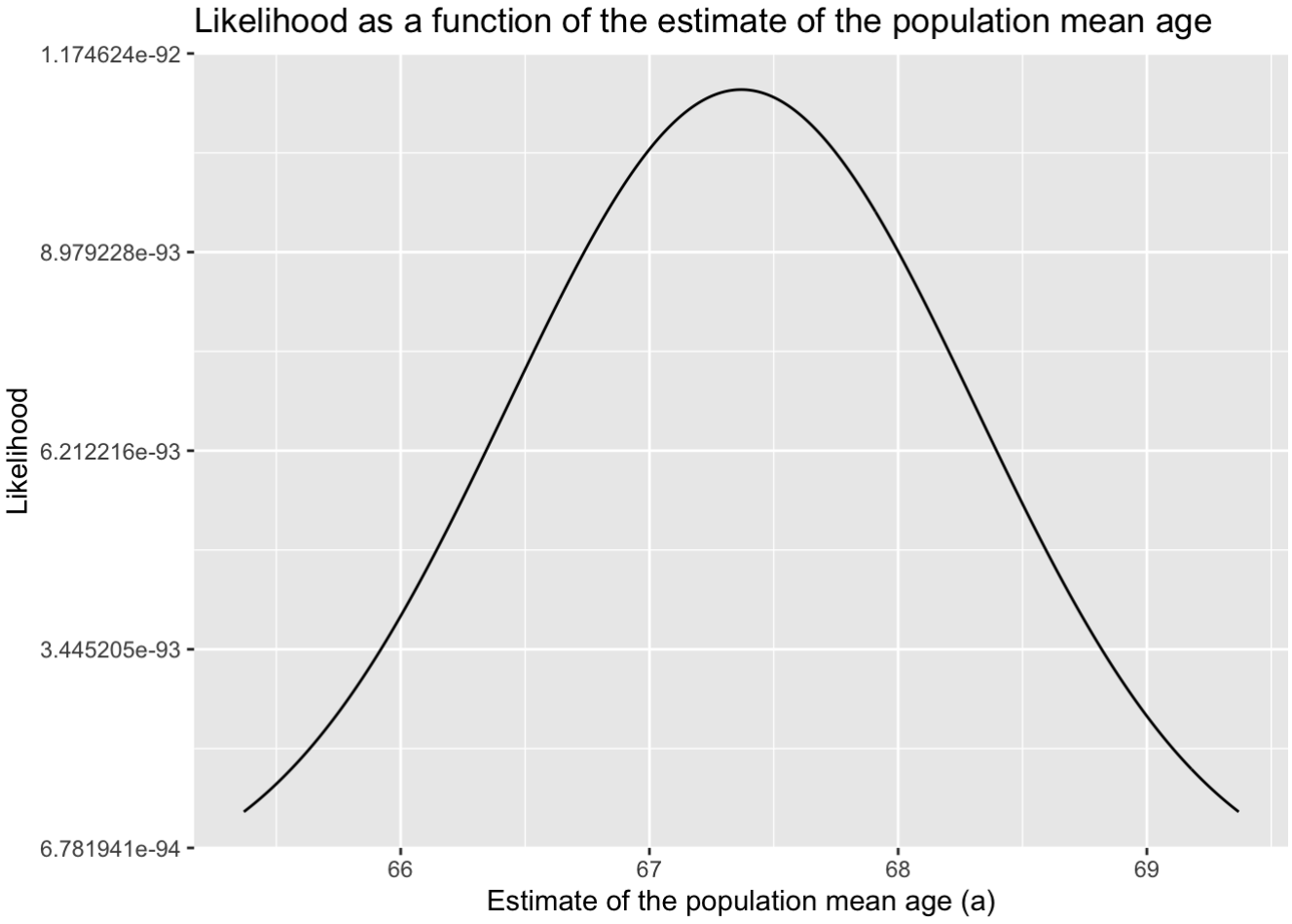
February 6, 2023

1. There is a dataset of clinical variables and self-reported fall history from people with Parkinson's disease available at [https://jlucasmckay.bmi.emory.edu/global/mckay_2021/S1.csv (https://jlucasmckay.bmi.emory.edu/global/mckay_2021/S1.csv)]. Identify the sample mean and sample standard deviation of the `Age` variable. Exclude any cases under 35 years old. **1 point**

```
data <- read.csv("https://jlucasmckay.bmi.emory.edu/global/mckay_2021/S1.csv")
data <- data[data$Age >= 35,]

m <- mean(data$Age)
s <- sd(data$Age)
l <- length(data$Age)
print(m)
## [1] 67.36935

print(s)
## [1] 7.418488
```

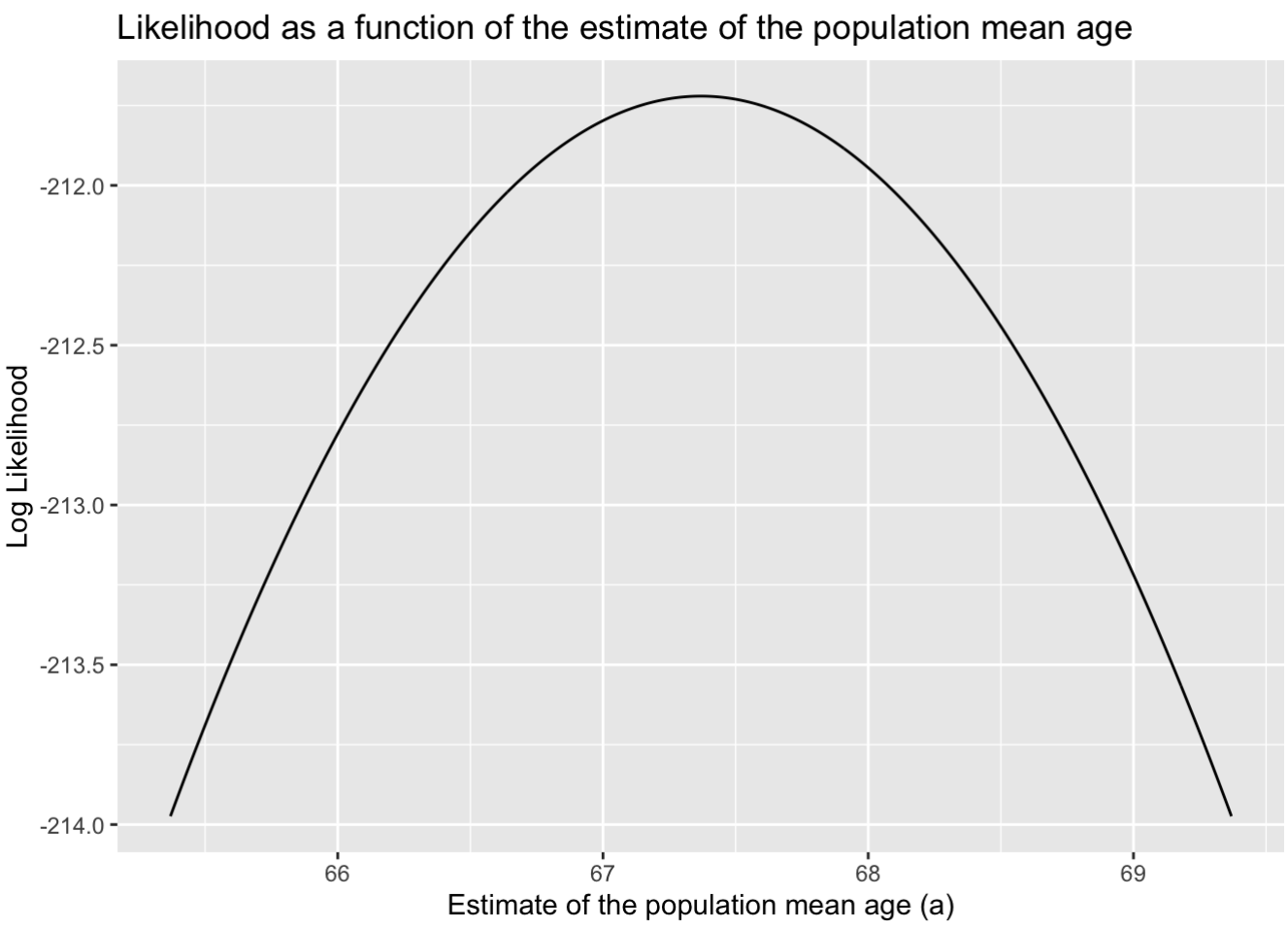
2. Calculate the likelihood of the estimates for the mean and standard deviation you have obtained given the sample. (Assume a normal distribution.) as you sweep the estimate of average age through ±2 years. Specifically, create a function `Lik(a)` that returns the likelihood of the data given an estimate of the population mean age `a` **(1/2 point)** and plot the likelihood as a function of `a` as you sweep it over ±2 years. **(1/2 point)**

```
# Define the likelihood function

Age <- data$Age
Lik <- function(a) {

  likelihood <- prod(dnorm(Age, mean = a, sd = s))
  return(likelihood)
}


library(ggplot2)

a_range <- seq(from = m - 2, to = m + 2, by = 0.01)
likelihood_values <- sapply(a_range, Lik)


ggplot(data.frame(a = a_range, likelihood = likelihood_values), aes(x = a, y = likelihoo
d)) +
  geom_line() +
  ggtitle("Likelihood as a function of the estimate of the population mean age") +
  xlab("Estimate of the population mean age (a)") +
  ylab("Likelihood")
```



3. Now plot the *log* likelihood as you sweep the estimate of average age through ±2 years. **(1 point)** *(Note that you have to use a sum, not a product, here.)*

```
Age <- data$Age
logLik <- function(a) {

  likelihood <- sum(dnorm(Age, mean = a, sd = s, log = TRUE))
  return(likelihood)
}


library(ggplot2)

a_range <- seq(from = m - 2, to = m + 2, by = 0.01)
log_likelihood_values <- sapply(a_range, logLik)


ggplot(data.frame(a = a_range, log_likelihood = log_likelihood_values), aes(x = a, y = l
og_likelihood)) +
  geom_line() +
  ggtitle("Likelihood as a function of the estimate of the population mean age") +
  xlab("Estimate of the population mean age (a)") +
  ylab("Log Likelihood")
```

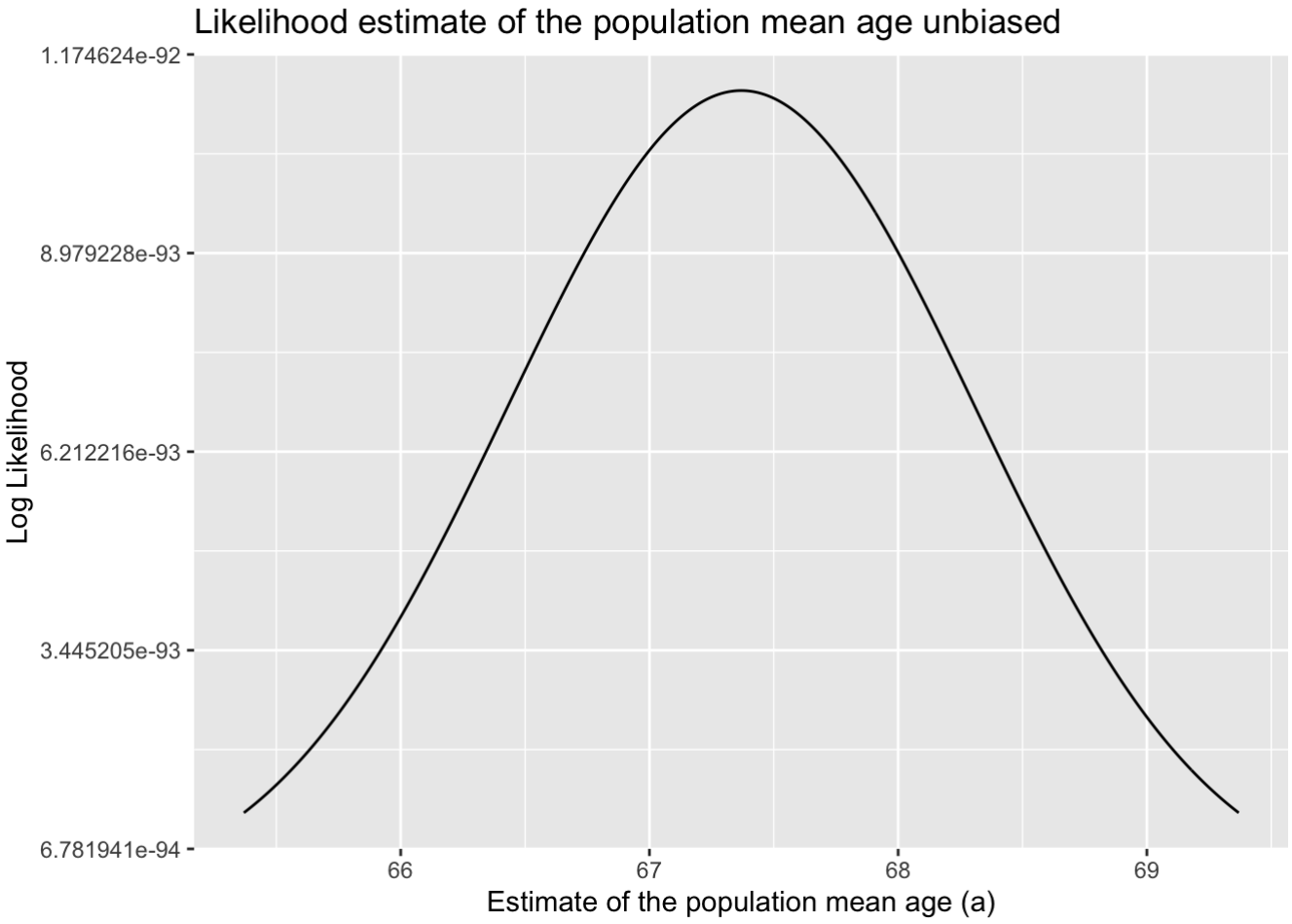


4. Using the answers for the last two questions, use a grid-based approach to find the maximum likelihood estimator for `a`. **(1/2 point)** Do the same for the maximum log-likelihood estimator. **(1/2 point)** Do they differ?

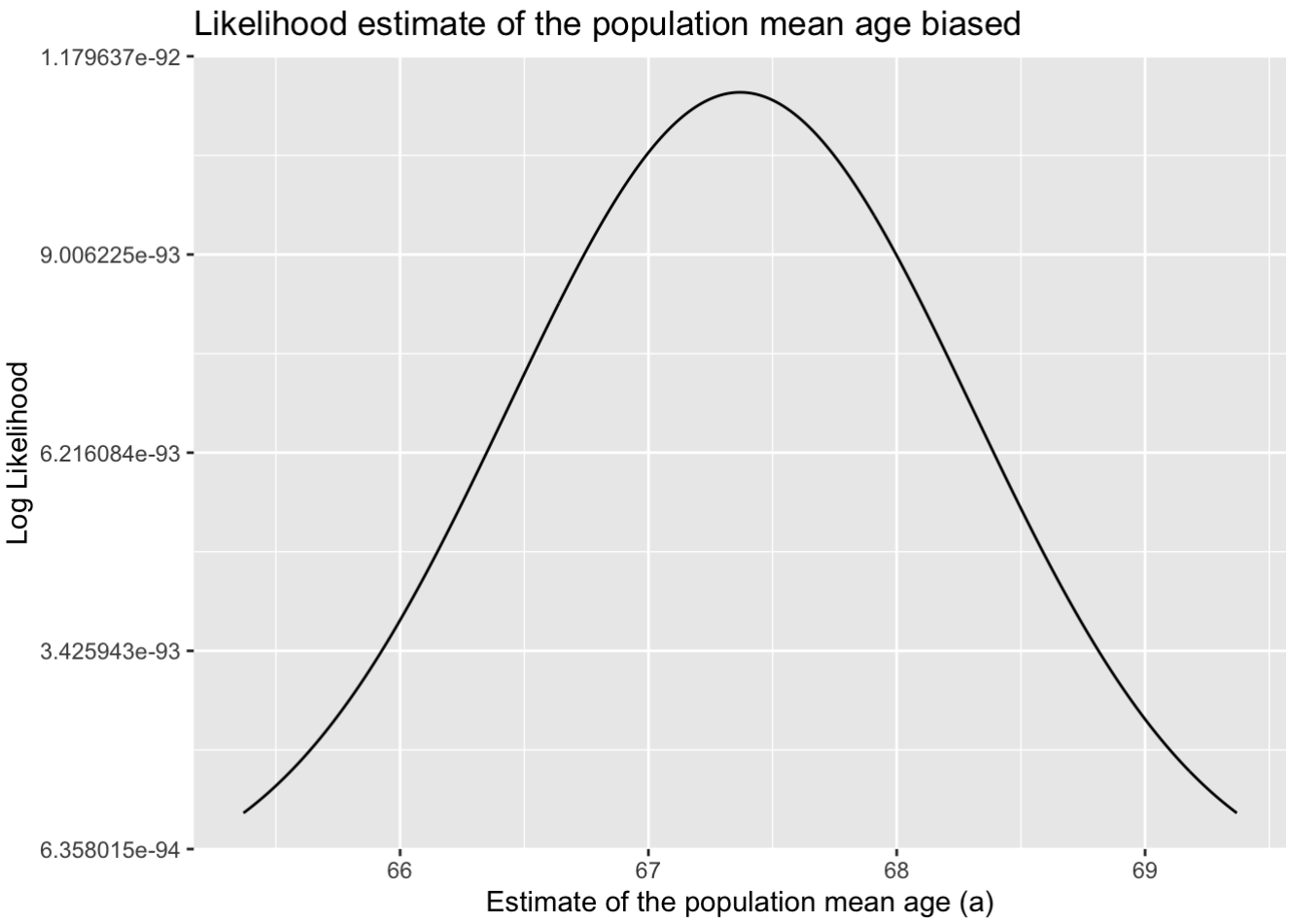
```
grid <- seq(from = min(Age), to = max(Age), by = 1)

#assigning a empty vector of length grid
lik_values <- numeric(length(grid))
loglik_values <- numeric(length(grid))

#calculating log likelihood and likelihood for all values in grid
for (i in 1:length(grid)) {
  lik_values[i] <- Lik(grid[i])
  loglik_values[i] <- logLik(grid[i])
}

#calculating max likelihood and log likelihood
max_index_likelihood <- which.max(lik_values)
max_index_loglikelihood <- which.max(loglik_values)

a_hat_likelihood <- grid[max_index_likelihood]
a_hat_loglikelihood <- grid[max_index_loglikelihood]


cat("liklihood age",a_hat_likelihood)
## liklihood age 67
cat("\nlog liklihood age", a_hat_loglikelihood)
##
## log liklihood age 67

#we see that they do not differ.
```

5. One of the things we noted was that using the sample standard deviation to estimate the population standard deviation can bias the estimate. Therefore, we often see `N-1` normalization in the standard deviation equation instead of `N`. Calculate and compare the likelihood of the data under the biased and unbiased estimators for the standard deviation. Which estimate for the standard deviation is larger? Which estimate is more likely? **(1 point)**

```
bias_sd <- sqrt(sum((data$Age - mean(data$Age))^2) / length(data$Age))
unbias_sd <- sqrt(sum((data$Age - mean(data$Age))^2) / (length(data$Age) - 1))


Lik_biased <- function(a) {
  likelihood <- prod(dnorm(Age, mean = a, sd = bias_sd))#sd(N)
  return(likelihood)
}


Lik_unbiased <- function(a) {
  likelihood <- prod(dnorm(Age, mean = a, sd = unbias_sd))#sd(N-1)
  return(likelihood)
}


a_range <- seq(from = m - 2, to = m + 2, by = 0.01)
likelihood_values <- sapply(a_range, Lik_unbiased)


ggplot(data.frame(a = a_range, likelihood = likelihood_values), aes(x = a, y = likelihoo
d)) +
  geom_line() +
  ggtitle("Likelihood estimate of the population mean age unbiased") +
  xlab("Estimate of the population mean age (a)") +
  ylab("Log Likelihood")
```




```
a_range <- seq(from = m - 2, to = m + 2, by = 0.01)
likelihood_values <- sapply(a_range, Lik_biased)


ggplot(data.frame(a = a_range, likelihood = likelihood_values), aes(x = a, y = likelihoo
d)) +
  geom_line() +
  ggtitle("Likelihood estimate of the population mean age biased") +
  xlab("Estimate of the population mean age (a)") +
  ylab("Log Likelihood")
```

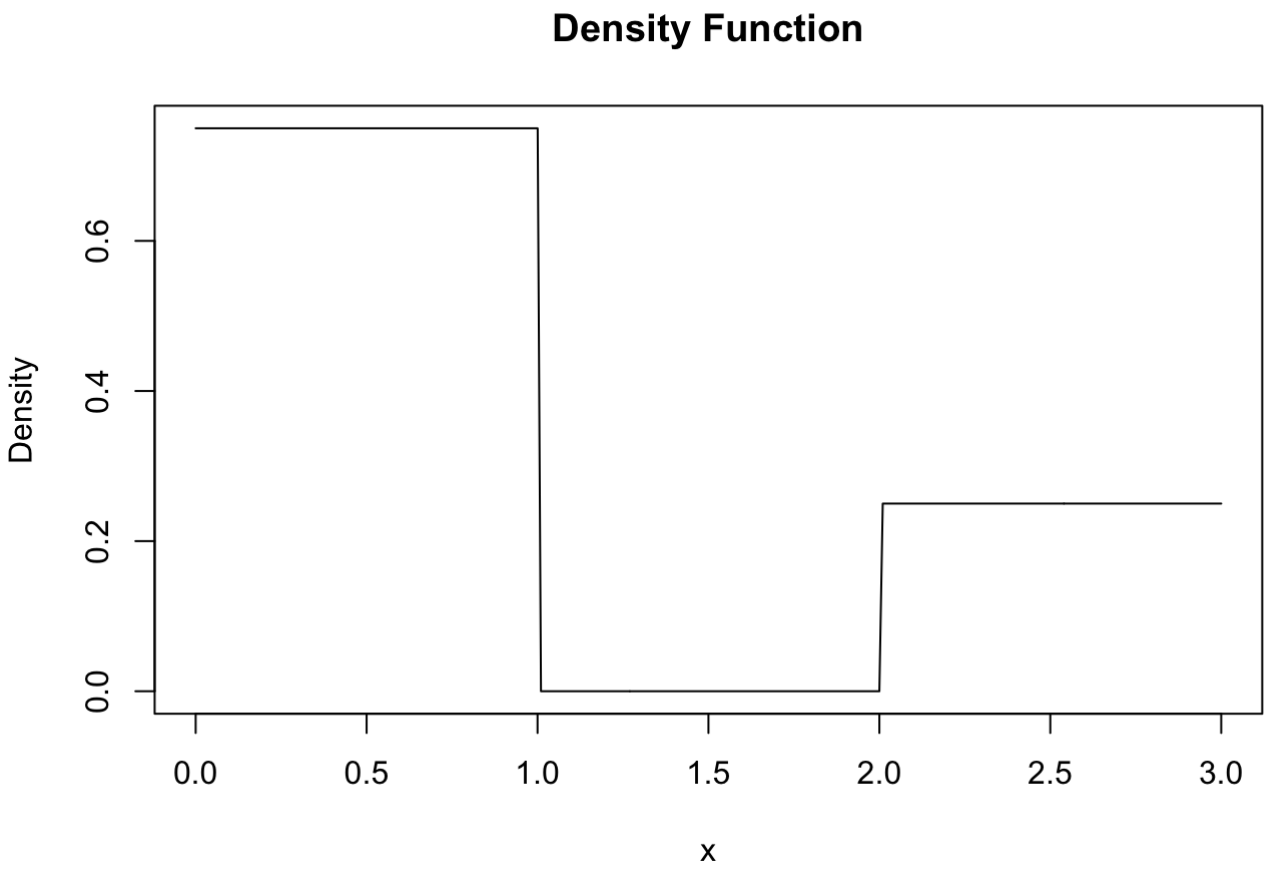



```
cat("Biased Estimate:", bias_sd, "\n")
## Biased Estimate: 7.358418
cat("Unbiased Estimate:", unbias_sd, "\n")
## Unbiased Estimate: 7.418488

#Unbiased estimate is larger than biased estimate

#Unbiased estimate is more likely as it represents real data more accurately when compar
ed to biased estimator
```

---

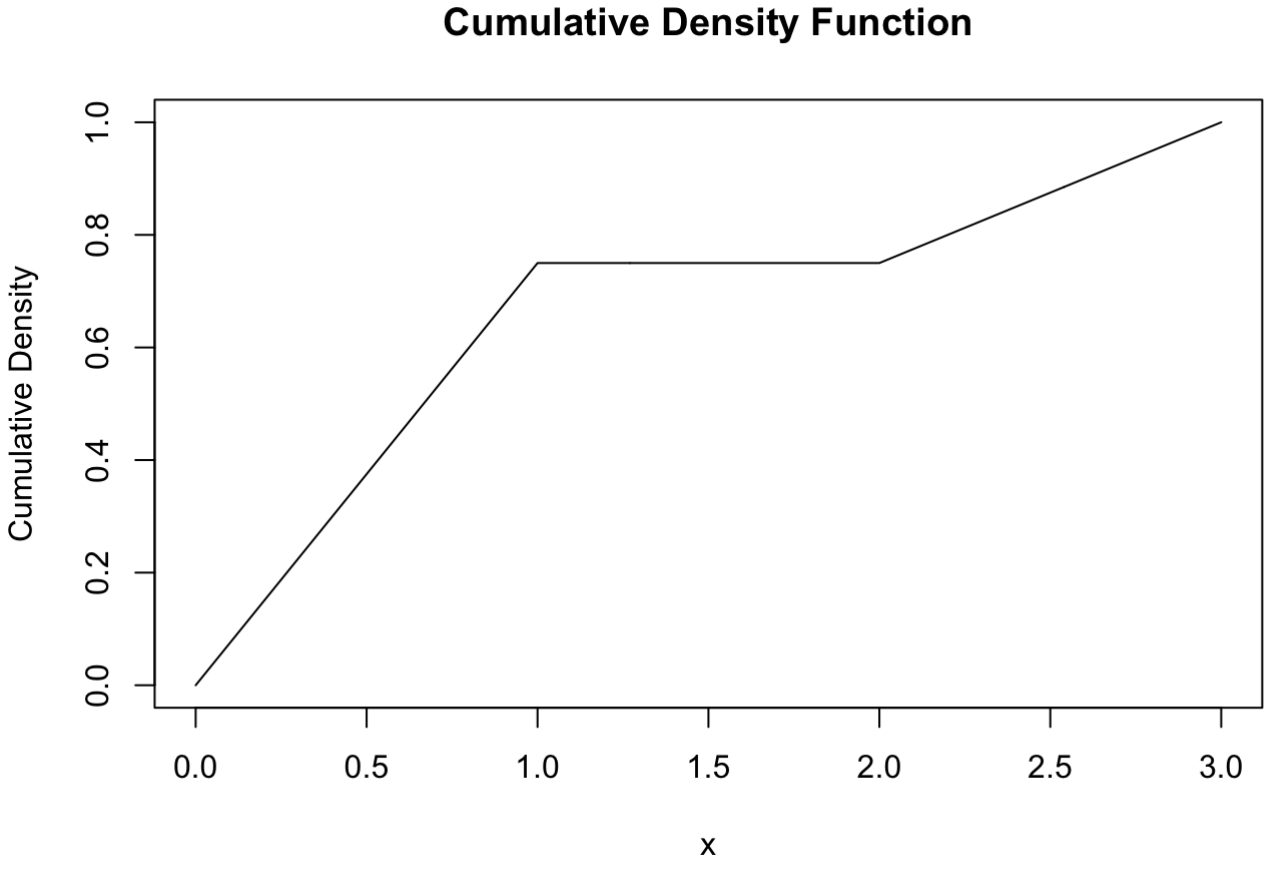
6. Let x be a continuous random variable with the piecewise-defined probability density function $f(x)$ equal to 0.75, $0 \leq x \leq 1$, 0.25, $2 \leq x \leq 3$, 0 elsewhere. Plot the density $f(x)$. **(1 point)**

```
library(ggplot2)

density_function <- function(x) {
  if (x >= 0 & x <= 1) {
    return (0.75)
  } else if (x > 1 & x <= 2) {
    return (0)
  } else if (x > 2 & x <= 3) {
    return (0.25)
  } else {
    return (0)
  }
}

x_values <- seq(from = 0, to = 3, by = 0.01)
density_values <- sapply(x_values, density_function)
plot(x_values, density_values, type = "l", xlab = "x", ylab = "Density", main = "Density
Function")
```



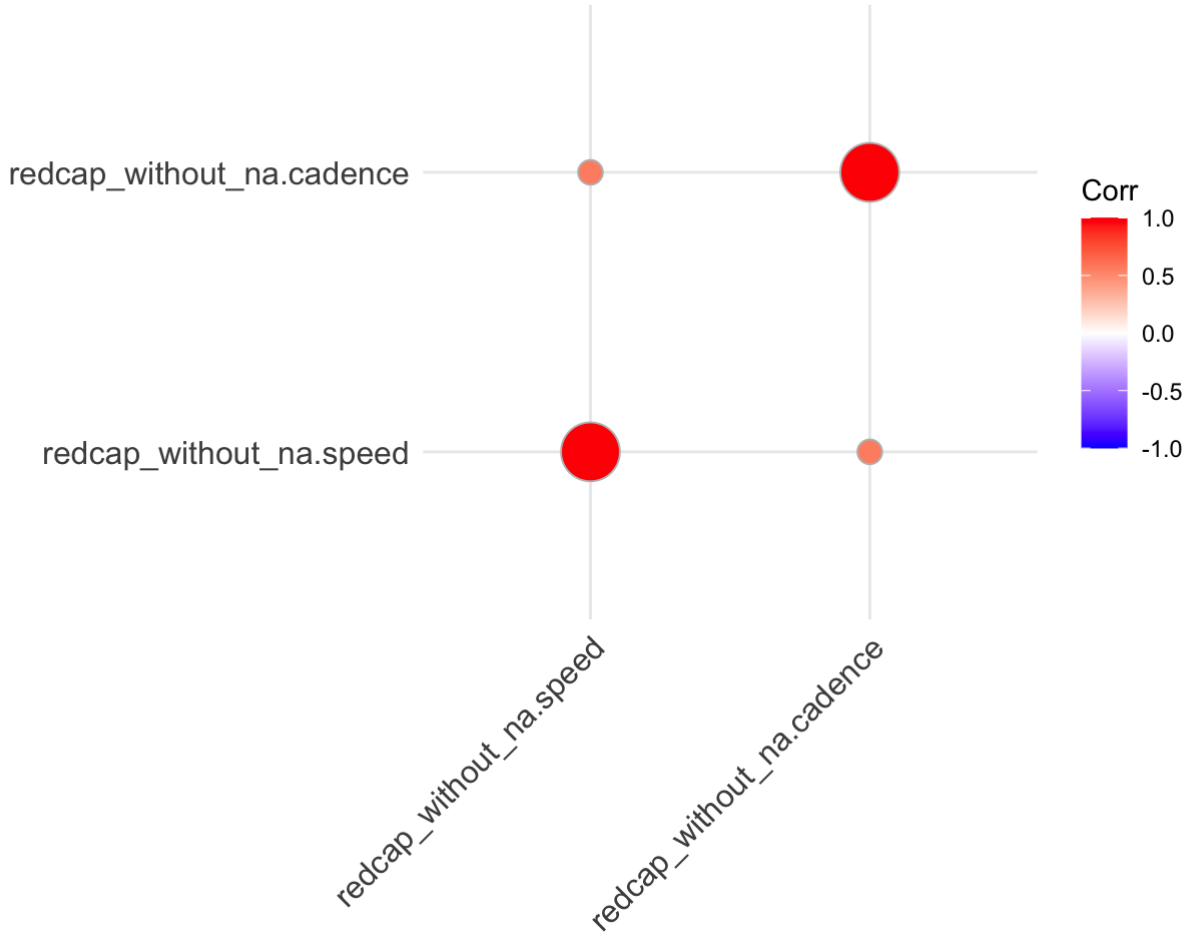
7. Plot the cumulative density $F(x)$. **(1 point)**

```
cumulative_density_function <- function(x) {
  if (x >= 0 & x <= 1) {
    return (0.75 * x)
  } else if (x > 1 & x <= 2) {
    return (0.75)
  } else if (x > 2 & x <= 3) {
    return (0.75 + 0.25 * (x - 2))
  } else {
    return (1)
  }
}

x_values <- seq(from = 0, to = 3, by = 0.01)
cumulative_density_values <- sapply(x_values, cumulative_density_function)
plot(x_values, cumulative_density_values, type = "l", xlab = "x", ylab = "Cumulative Den
sity", main = "Cumulative Density Function")
```

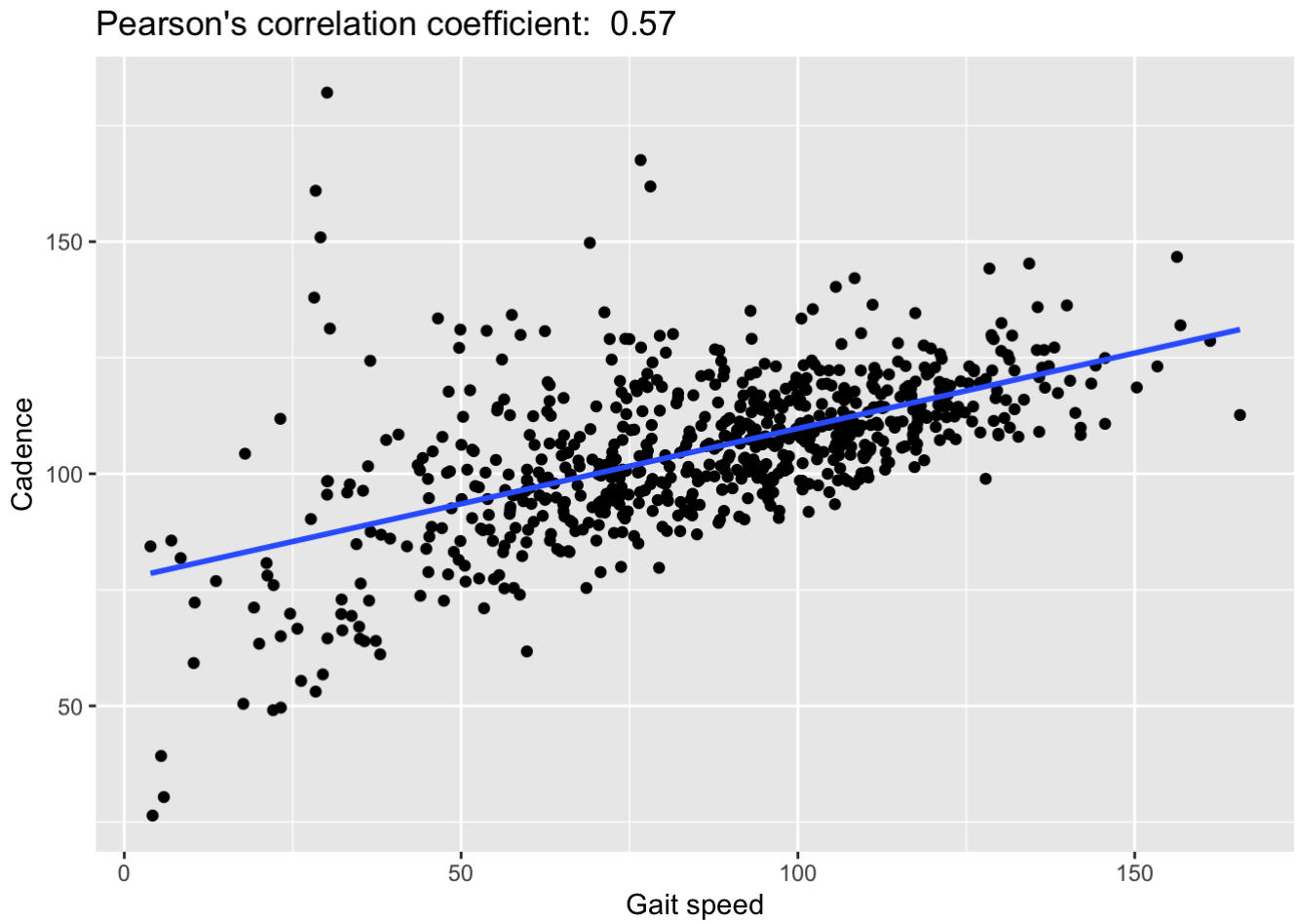


8. Using the sample redcap dataset available at [https://jlucasmckay.bmi.emory.edu/global/bmi510/gait.csv (https://jlucasmckay.bmi.emory.edu/global/bmi510/gait.csv)], identify the unique patients and summarize the ratio of women to men. Also report how many missing values there are. (The sex variable codes 0 for male, 1 for female.) **(1/2 point)** Then, calculate (and plot) the (Pearson's) correlation between gait speed and cadence. **(1/2 point)**

```
library(dplyr)
library(ggcorrplot)
library(corrplot)
library(ggcorrplot)



redcap <- read.csv("https://jlucasmckay.bmi.emory.edu/global/bmi510/gait.csv")
unique_patients <- unique(redcap$record_id)

cat("No of Unique patients are : ",length(unique_patients))
## No of Unique patients are :  935

#we need to filter out the rows where the reocrd_id reappears before calculating the rat
io of men to women
redcap_unique <- redcap %>% filter(!duplicated(record_id))

m <- redcap_unique$sex == 0
f <- redcap_unique$sex == 1

men <- length(m[m == TRUE])
women <- length(f[f == TRUE])

ratio_of_men_women <- men/women

cat("\nRatio of men to women :",ratio_of_men_women)
##
## Ratio of men to women : 1.301811

missing_vals <- is.na(redcap_unique$sex)
missing_values <- length(missing_vals[missing_vals == TRUE])

cat("\nThe sex column has ",missing_values," missing values\n")
##
## The sex column has  209  missing values

redcap_without_na <- redcap %>% filter(!is.na(speed) & !is.na(cadence))

df_x <- data.frame(redcap_without_na$speed)

df_y <- data.frame(redcap_without_na$cadence)


correlation <- cor(df_x, df_y)
correlation_matrix <- cor(cbind(df_x, df_y))

ggcorrplot(correlation_matrix, method = "circle")
```




```
#df_x <- data.frame(x_l <- redcap_without_na$speed, lab <- rep("gait speed", nrow(df_
x)))

#df_y <- data.frame(x_l <- redcap_without_na$cadence, lab <- rep("cadence", nrow(df_y)))
#colnames(df_x) <- c("x_l", "lab")
#colnames(df_y) <- c("x_l", "lab")

#redcap_without_na_combined <- bind_rows(df_x,df_y)


ggplot(redcap_without_na, aes(x = speed, y = cadence)) +
geom_point() +
geom_smooth(method="lm", se=FALSE)+
ggtitle(paste("Pearson's correlation coefficient: ", round(correlation, 2))) +
xlab("Gait speed") +
ylab("Cadence")
```



9. We have not discussed (at least at length) joint frequency functions / probability mass functions, but they are a generalization of probability mass functions for a single random variable. Say the joint frequency function of two discrete random variables, X and Y, is as follows:

X	Y=1	Y=2	Y=3	Y=4
1	0.10	0.05	0.02	0.02
2	0.05	0.20	0.05	0.02
3	0.02	0.05	0.20	0.04
4	0.02	0.02	0.04	0.10

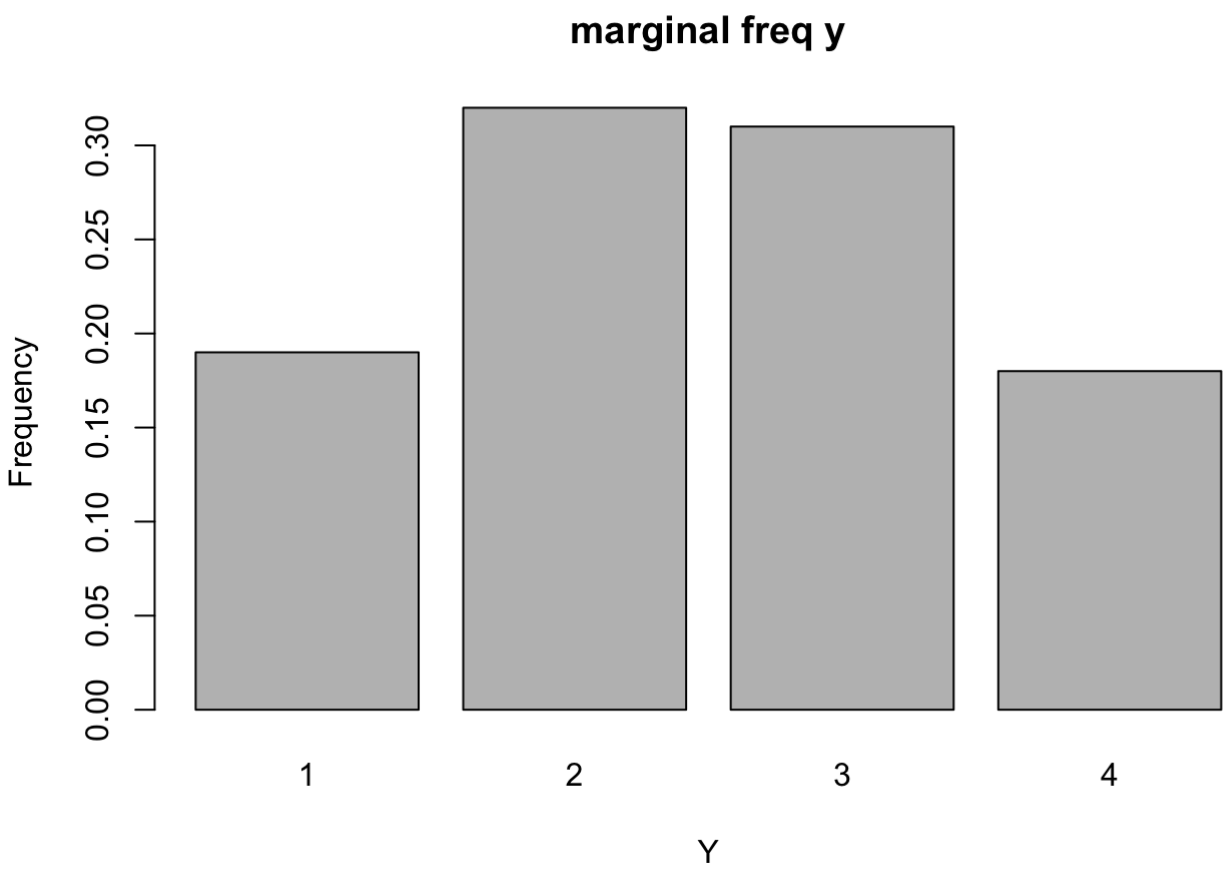
These data are available at [https://jlucasmckay.bmi.emory.edu/global/bmi510/joint_frequency.csv (https://jlucasmckay.bmi.emory.edu/global/bmi510/joint_frequency.csv)] Columns 2 through 4 are named Y1, Y2, Y3, and Y4. Completely melt the data **(1/2 point)** and extract numerical values from the codes `Y1`, `Y2`, etc. **(1/2 point)**

```
library(tidyr)
library(reshape2)


data <- read.csv("https://jlucasmckay.bmi.emory.edu/global/bmi510/joint_frequency.csv")
melted_data <- melt(data, id.vars="X")

print(melted_data$value)
##  [1] 0.10 0.05 0.02 0.02 0.05 0.20 0.05 0.02 0.02 0.05 0.20 0.04 0.02 0.02 0.04
## [16] 0.10
```

10. Find and plot the marginal frequency functions of X and Y. In the two-by-two table above, these would be the row and column sums. (The margins.) To what do the frequencies sum? **(1 point)**

```
marginal_frequency_X <- c(0.10 + 0.05 + 0.02 + 0.02,
                          0.05 + 0.20 + 0.05 + 0.02,
                          0.02 + 0.05 + 0.20 + 0.04,
                          0.02 + 0.02 + 0.04 + 0.10)

marginal_frequency_Y <- c(0.10 + 0.05 + 0.02 + 0.02,
                          0.05 + 0.20 + 0.05 + 0.02,
                          0.02 + 0.05 + 0.20 + 0.04,
                          0.02 + 0.02 + 0.04 + 0.10)

cat("Marg freq x:", marginal_frequency_X, "\nMarg freq y",marginal_frequency_Y)
## Marg freq x: 0.19 0.32 0.31 0.18
## Marg freq y 0.19 0.32 0.31 0.18

barplot(marginal_frequency_X, names.arg=c(1,2,3,4), xlab="X", ylab="Frequency", main =
"marginal freq x")
```


```
barplot(marginal_frequency_Y, names.arg=c(1,2,3,4), xlab="Y", ylab="Frequency", main =
"marginal freq y")
```




```
#The frequencies represent the distribution of the discrete random variables X and Y. Th
e sum of the marginal frequencies represent the total count of occurrences of X and Y.
```