Lab 3

Kristina Arevalo

Mon Sep 28 2020

Contents

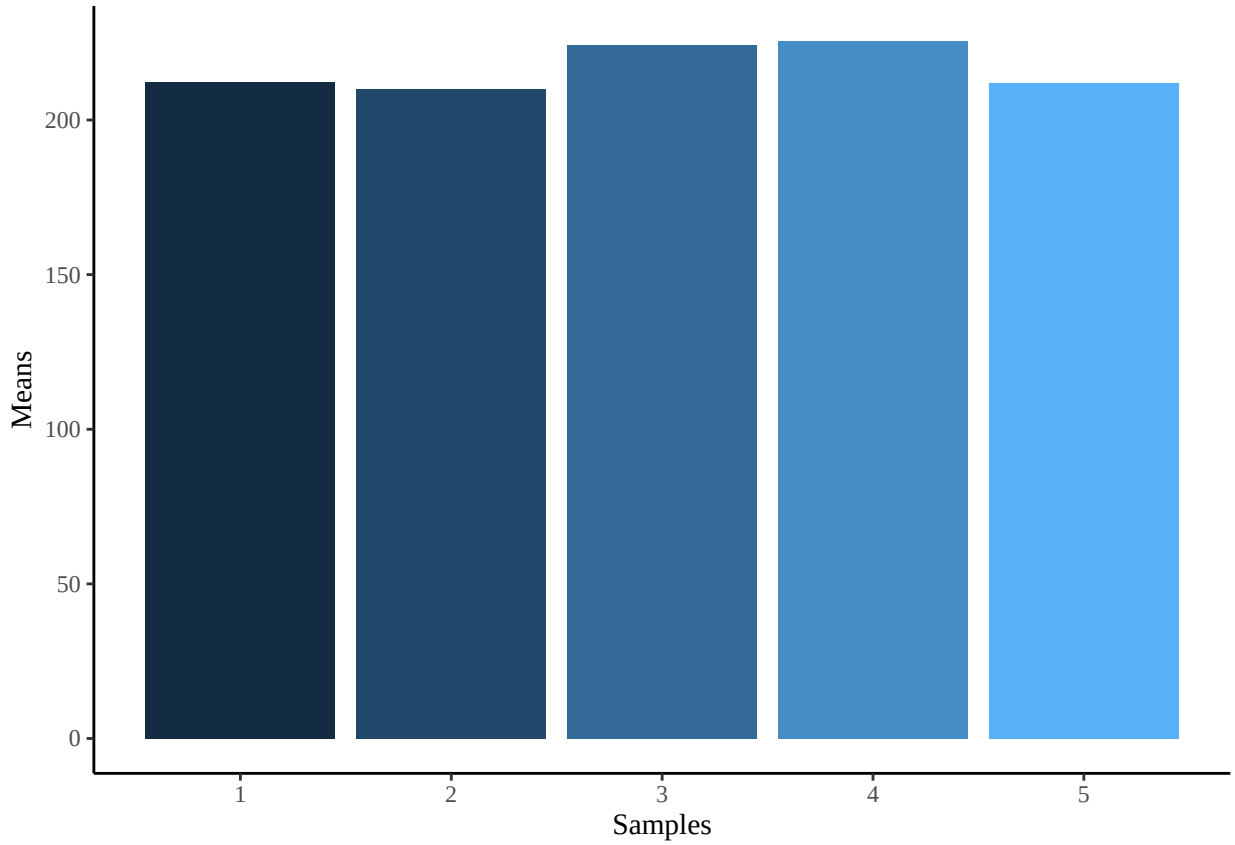
Problem 1

Create five samples of 25 observations from a normal distribution with mean 200, and standard deviation 100. Compute the mean of each sample, and plot the means in a graph using ggplot2. (1 point)

```
observations <- rnorm(5*25, 200, 100)
samples <- rep(1:5, each = 25)
my_data <- data.frame(samples, observations)

means <- my_data %>%
  group_by(samples) %>%
  summarize(mean=mean(observations))

sample_means_plot<- ggplot(means, aes(samples, mean, fill = samples)) +
  geom_bar(stat = "identity") +
  theme_classic()+
  theme(text = element_text(family = "Times"), legend.position = "none") +
  labs(x = "Samples", y = "Means")
sample_means_plot
```



Confidence = 100

Problem 2

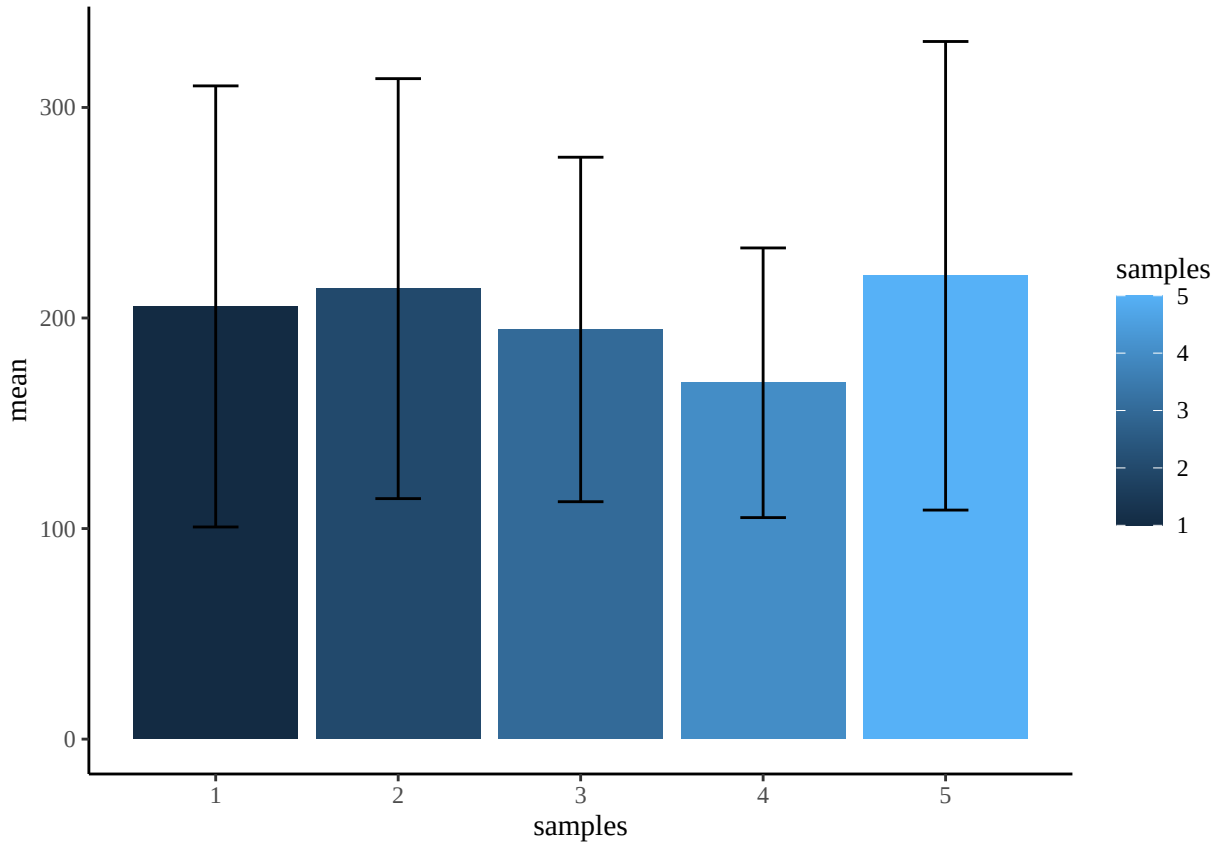
Additionally calculate the standard deviation of each sample from above. Use the standard deviations for error bars, and produce another graph with the means along with error bars using ggplot2. (1 point)

```
observations <- rnorm(5*25, 200, 100)
samples <- rep(1:5, each = 25)
my_data <- data.frame(samples, observations)

means <- my_data %>%
  group_by(samples) %>%
  summarize(mean=mean(observations), sd= sd(observations))

error_graph <- ggplot(means, aes(samples, mean, fill= samples)) +
  geom_bar(stat="identity") +
  geom_errorbar(aes(ymin = mean - sd,
                    ymax = mean + sd),
                width = .25)+
  theme_classic() +
  theme(text = element_text(family = "Times"))

error_graph
```



confidence = 100

Problem 3

Demonstrate that the sample mean across a range of n, is an unbiased estimator of the population mean using a monte-carlo simulation. (2 points). The population is a normal distribution with mean = 10, standard deviation = 5. Test a variety of n (sample size), including n = 2, 5, 10, 50, and 100 For each sample size n, your task is to draw 10,000 samples of that size, then for each sample, calculate the sample mean. If the mean is unbiased, then we expect that "on average" the sample means will be the same as the population mean. To determine if this is true, compute the mean of the sample means that you produce to see if it is close to the population mean. Show the mean of the sample means for each sample size.

```
sample_sizes <- c(2,5,10,50,100)

for(n in sample_sizes){
  observations <- rnorm(n*10000, 10, 5)
  samples <- rep(1:10000, each = n)
  my_data <- data.frame(samples,observations)

  summarized_data <- my_data %>%
    group_by(samples) %>%
    summarize(sample_means = mean(observations))
}

print(mean(summarized_data$sample_means))
```

```
## [1] 9.988709
```

Confidence = 50 ; did it the long way and then checked the video and saw you did a for loop and changed my code , however even when I use the same code it only prints out 1 mean not sure why.

Problem 4

Use a monte carlo simulation to compare the standard deviation formulas (divide by N vs N-1), and show that the N-1 formula is a better unbiased estimate of the population standard deviation, especially for small n. (2 points) Use the same normal distribution and samples sizes from above Rather than computing the mean for each sample, compute both forms of the standard deviation formula, including the sample standard deviation that divides by N-1, and the regular standard deviation that divides by N You should have 10,000 samples for each sample size, and 10,000 standard deviations for each the sample and regular standard deviation. Your task is to find the average of each, for each sample-size.

```
sd_N <- function(x){
  sqrt(sum((mean(x)-x)^2) / length(x))
}

sample_sizes <- c(2,5,10,50,100)
sim_sample_means <- c()
sim_sample_sd <- c()
sim_sample_sd_N <- c()
for(i in length(sample_sizes)){
  observations <- rnorm(sample_sizes[i]*10000, 10, 5)
  samples <- rep(1:10000, each = sample_sizes[i])
  my_data <- data.frame(samples,observations)
```

```
  summarized_data <- my_data %>%
    group_by(samples) %>%
    summarize(sample_sd = sd(observations),
              sample_sd_N= sd_N(observations))
  sim_sample_sd[i] <- mean(summarized_data$sample_sd)
  sim_sample_sd_N[i] <- mean(summarized_data$sample_sd_N)
}

sim_data <- data.frame(n = rep(sample_sizes, 2),
                       est = c(sim_sample_sd_N,sim_sample_sd),
                       formula = c(rep("N", 5), rep("N-1", 5)))

sim_data
```

```
##      n      est formula
## 1    2       NA       N
## 2    5       NA       N
## 3   10       NA       N
## 4   50       NA       N
## 5  100 4.962922       N
## 6    2       NA     N-1
## 7    5       NA     N-1
## 8   10       NA     N-1
## 9   50       NA     N-1
## 10 100 4.987925     N-1
```

confidence= 0 ; code still doesn't work for me not sure why

Which of the standard deviations is more systematically biased? That is, which one is systematically worse at estimating the population standard deviation?

Dividing by just N is more biased