Homework 3 Solution

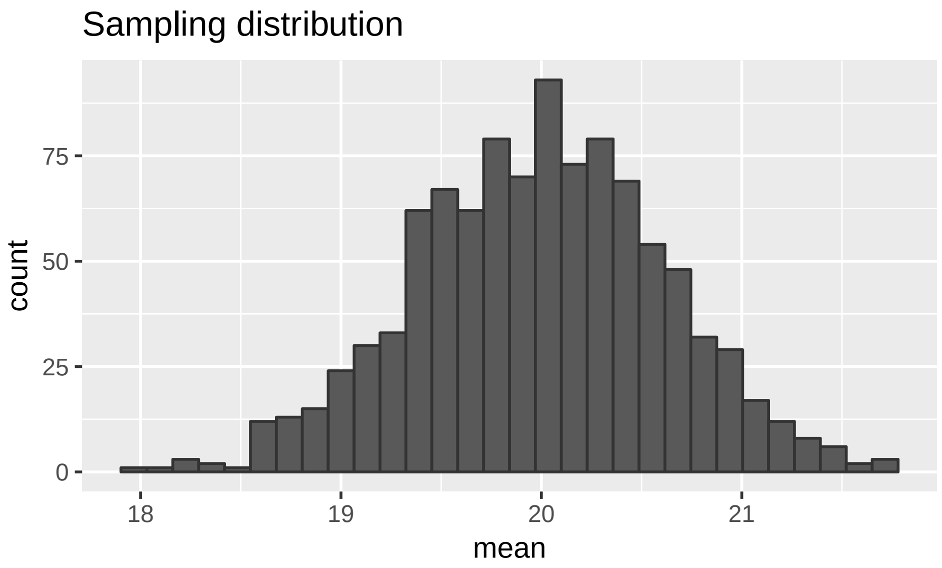
Math 445, Spring 2017

Exercise 8

Consider a population that has a gamma distribution with parameters $r = 5$, $\lambda = 1/4$.

Part a. In order to simulate an approximate sampling distribution we need to repeatedly obtain random samples of size $n = 200$ from the population distribution and calculate the sample mean.

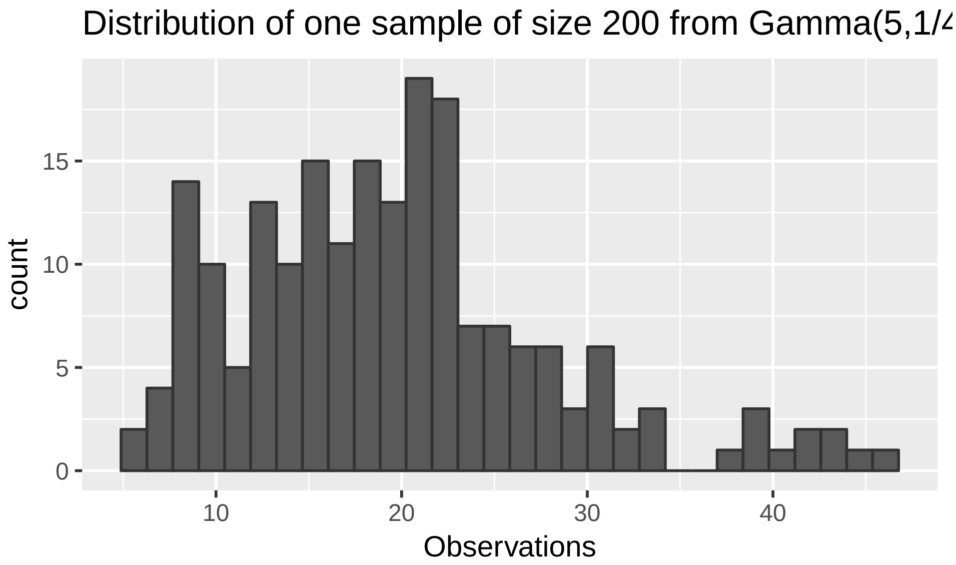
```
# Construct the sampling distribution
N <- 1000
sampling_dsn <- numeric(N)
for(i in 1:N) {
 samp <- rgamma(200, 5, 1/4)
 sampling_dsn[i] <- mean(samp)
}

# Plot it
ggplot(data = data.frame(mean = sampling_dsn), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20") +
  ggtitle("Sampling distribution")
```



```
# Summarize it
mean_samp_dsn <- mean(sampling_dsn); mean_samp_dsn

## [1] 19.98865

sd_samp_dsn <- sd(sampling_dsn); sd_samp_dsn

## [1] 0.6215815
```

Part b.

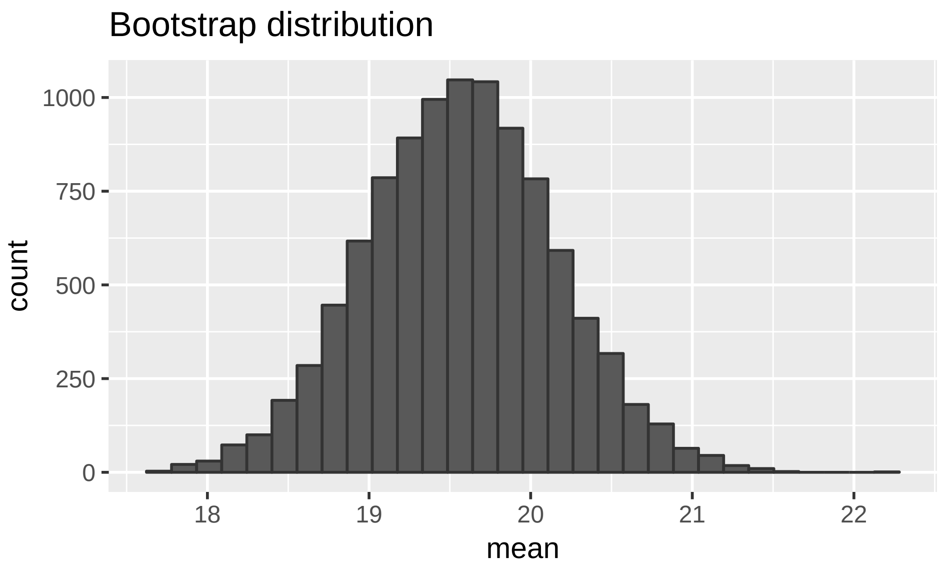
```
# Draw one sample of size 200
samp5.8b <- rgamma(200, 5, 1/4)

# Plot it
ggplot(data = data.frame(obs = samp5.8b), mapping = aes(x = obs)) +
  geom_histogram(colour = "gray20") +
  xlab("Observations") +
  ggtitle("Distribution of one sample of size 200 from Gamma(5,1/4)")
```



```
# Summarize it
mean_sample <- mean(samp5.8b); mean_sample

## [1] 19.55945

sd_sample <- sd(samp5.8b); sd_sample

## [1] 8.352162
```

Part c. Using the sample from part b, we can obtain the bootstrap distribution in the usual way:

```
# Resample with replacement from the sample
N <- 10000
boot_means <- numeric(N)
for(i in 1:N) {
  boot_means[i] <- mean(sample(samp5.8b, replace = TRUE))
}

# Plot it
ggplot(data = data.frame(mean = boot_means), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20") +
  ggtitle("Bootstrap distribution")
```



```
# Summarize it
mean_boot <- mean(boot_means); mean_boot

## [1] 19.56836

sd_boot <- sd(boot_means); sd_boot

## [1] 0.5895704
```

Part d. Compare the bootstrap distribution to the approximate theoretical sampling distribution by creating a table like Table 5.2.

Distribution	Mean	SD
Popualation	20.00	8.94
Sampling distribution	19.99	0.62
Sample	19.56	8.35
Bootstrap distribution	19.57	0.59

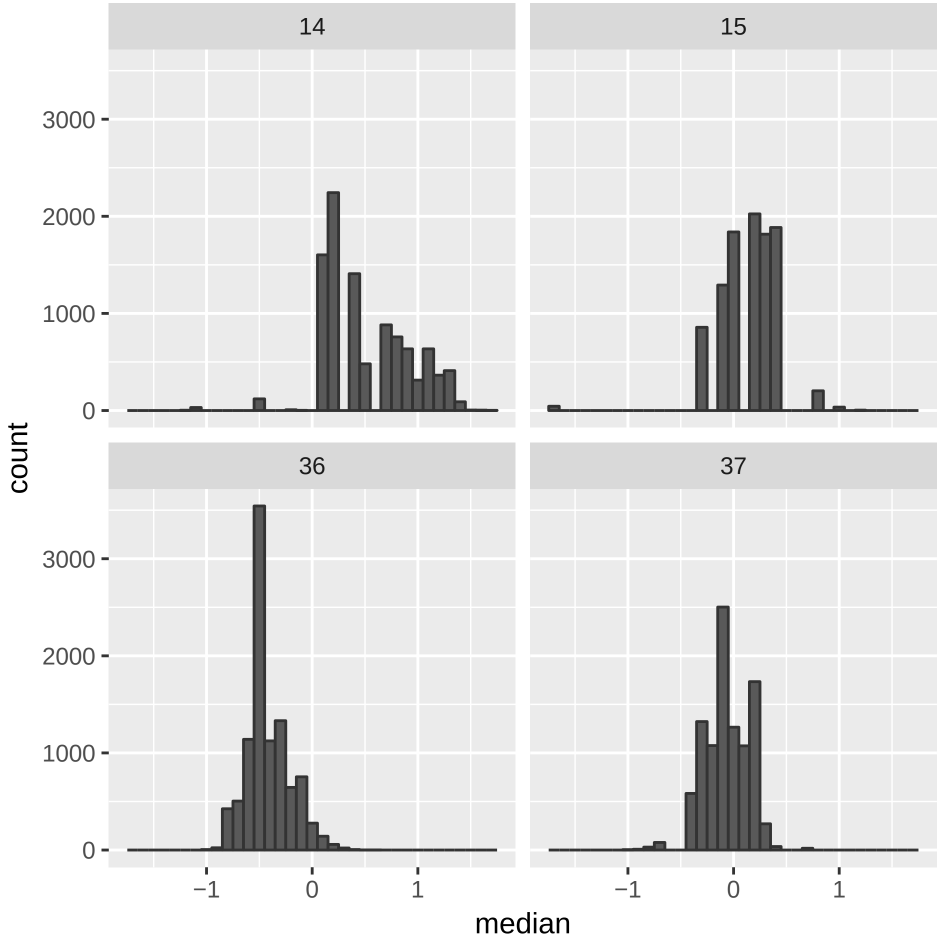
Part e.

Changing the code to reflect the different sample sizes ($n = 50$ and $n = 10$) and comparing all of the results you find that the standard error increases as sample size decreases.

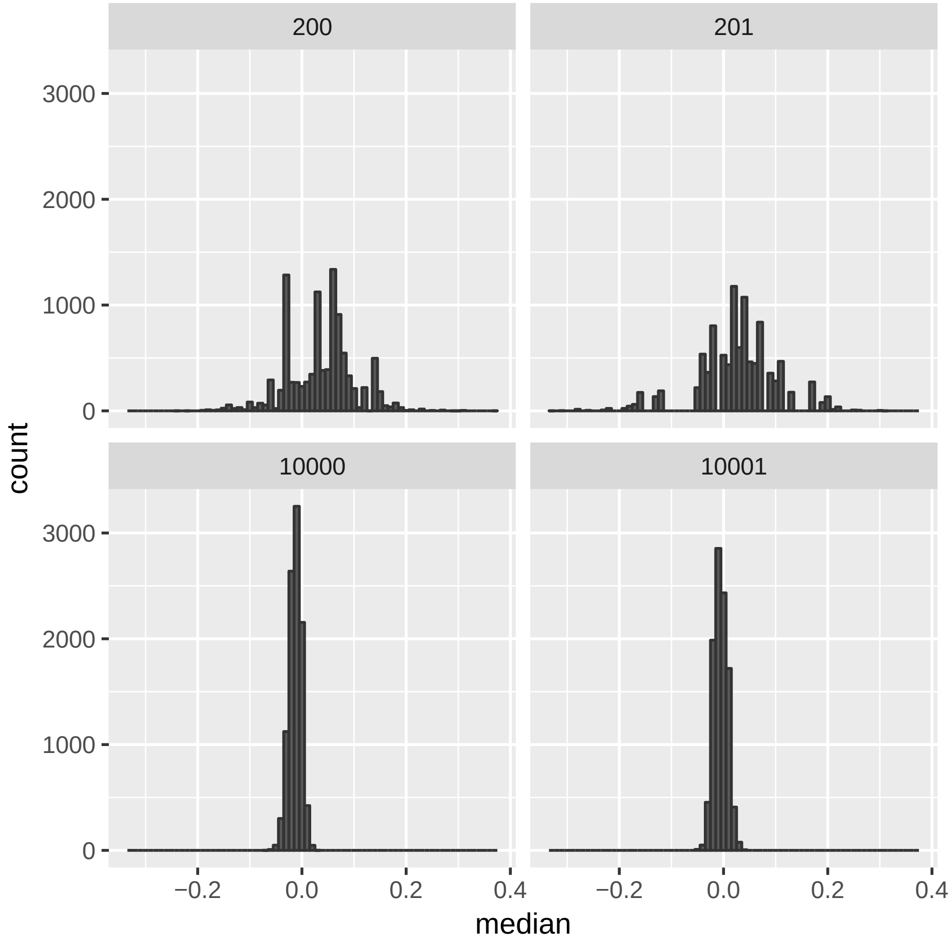
Exercise 9

Below is the code I used to generate all of the simulation settings. Notice that I used nested for loops. I didn't expect this, but it is easier than replicating this all by hand!

```
even_samp_sizes <- c(14, 36, 200, 10^4)
res5.9 <- data.frame(n = NULL, median = NULL) # inefficient, but quick
for(so in even_samp_sizes)
{
  ne <- so      # n even
  no <- so + 1 # n odd

  wwe <- rnorm(ne)
  wwo <- rnorm(no)

  N <- 10^4
  even.boot <- numeric(N)
  odd.boot <- numeric(N)
  for(i in 1:N)
  {
    x.even <- sample(wwe, ne, replace = TRUE)
    x.odd <- sample(wwo, no, replace = TRUE)
    even.boot[i] <- median(x.even)
    odd.boot[i] <- median(x.odd)
  }
  res5.9 <- rbind(res5.9, cbind(n = ne, median = even.boot), cbind(n = no, median = odd.boot))
}

# Make the plots
library(dplyr)
ggplot(data = filter(res5.9, n < 200), mapping = aes(x = median)) +
  geom_histogram(colour = "gray20", binwidth = .1) +
  facet_wrap(~n, ncol = 2)
```



```
ggplot(data = filter(res5.9, n >= 200), mapping = aes(x = median)) +
  geom_histogram(colour = "gray20", binwidth = .01) +
  facet_wrap(~n, ncol = 2)
```

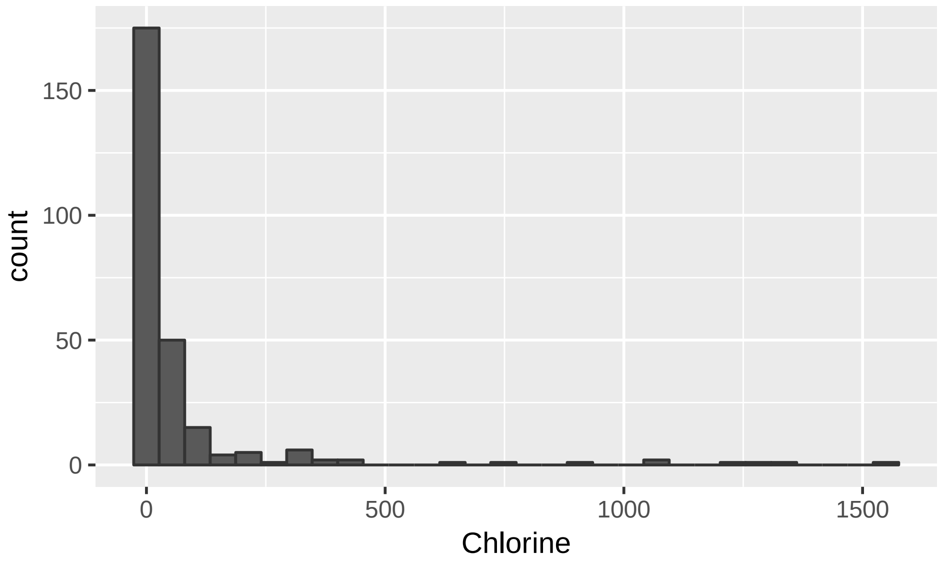


Aside from the fact that it is difficult to pick a uniform bin width for these plots, we find that for small n, the sampling distribution for the median is quite "granular" when n is odd. This is because the median will be one of the sample value in this case. As n increases, this granularity becomes less apparent.

Exercise 10

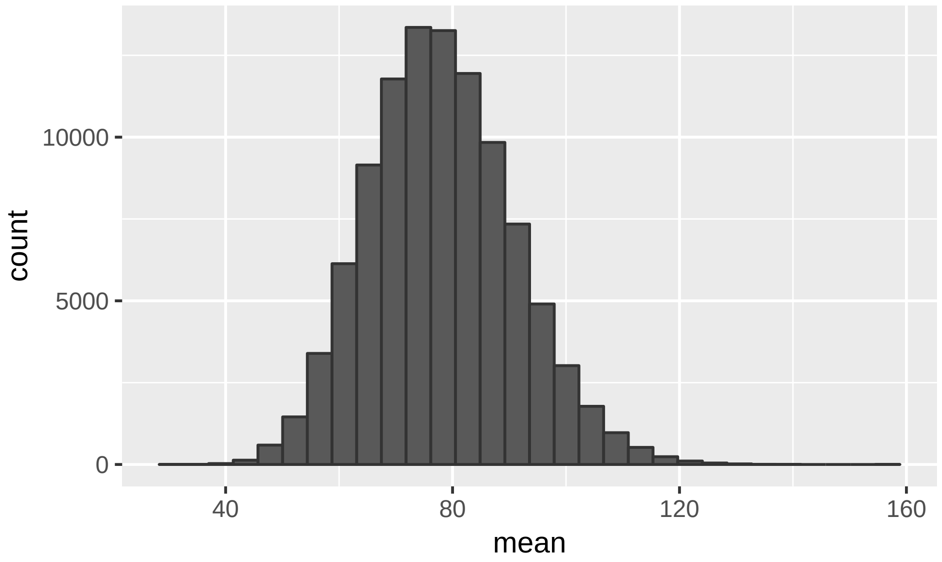
Part a. The distribution of chlorine concentrations is strongly skewed to the right. The five number summary is given in the below R code. Additionally, we see that there are two missing values.

```
library(resampledata)


summary(Bangladesh$Chlorine)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    1.00    5.00   14.20   78.08   55.50 1550.00       2

ggplot(data = Bangladesh, mapping = aes(x = Chlorine)) +
  geom_histogram(colour = "gray20")
```



Part b. Note: Before bootstrapping, we should remove the missing values.

```
chlorine <- subset(Bangladesh, select = Chlorine, subset = !is.na(Chlorine), drop = TRUE)

N <- 10^5
chlorine_boot <- numeric(N)
for(i in 1:N)
{
  x <- sample(chlorine, replace = TRUE)
  chlorine_boot[i] <- mean(x)
}

ggplot(data = data.frame(mean = chlorine_boot), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20")
```



```
# boot mean
mean(chlorine_boot)

## [1] 78.06473

# boot SE
sd(chlorine_boot)
```

```
## [1] 12.79908
```

We see that the bootstrap distribution of the mean is unimodal and symmetric.

Part c.

```
ci5.10 <- quantile(chlorine_boot, probs = c(0.025, 0.975))
ci5.10

##      2.5%     97.5%
##  54.88549 104.80670
```

We are 95% confident that the average chlorine concentration for wells in Bangladesh is between 54.89 and 104.81.

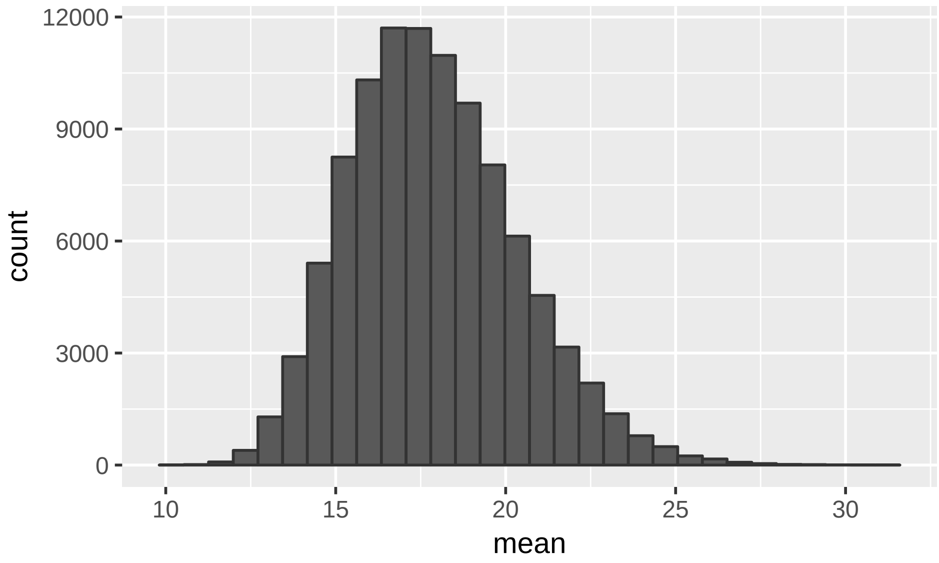
Part d.

```
# bias
bias5.10 <- mean(chlorine_boot) - mean(chlorine)
bias5.10

## [1] -0.0192821

# SE
se5.10 <- sd(chlorine_boot)
se5.10

## [1] 12.79908
```

The bootstrap estimate of bias (for my simulations) is -0.019. Further, we find that $bias/SE \approx -0.002$; thus, the bias is approximately 0.2% of the standard error.

Exercise 11

```
N <- 10^5
trim_boot <- numeric(N)
for(i in 1:N)
{
  x <- sample(chlorine, replace = TRUE)
  trim_boot[i] <- mean(x, trim = 0.25)
}

ggplot(data = data.frame(mean = trim_boot), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20")
```



```
# bias
bias5.11 <- mean(trim_boot) - mean(chlorine, trim = 0.25)
bias5.11

## [1] 0.2429371

# SE
se5.11 <- sd(trim_boot)
se5.11

## [1] 2.464764

# bias/SE
bias5.11/se5.11

## [1] 0.09856403

# CI
ci5.11 <- quantile(trim_boot, probs = c(0.025, 0.975))
ci5.11

##     2.5%    97.5%
## 13.66517 23.21926
```

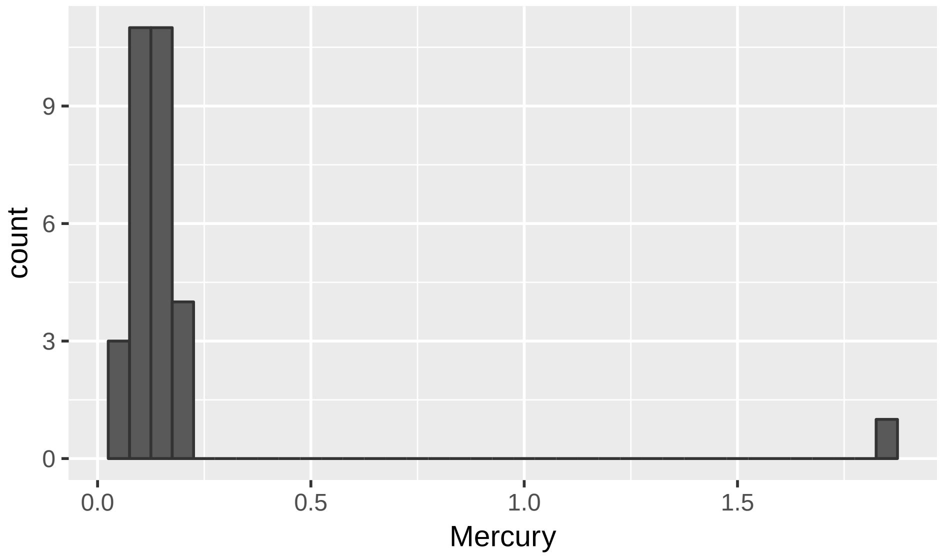
We see that the bootstrap distribution for the trimmed mean is slightly skewed to the right. Further we see that the bootstrap distribution for the trimmed mean is narrower (i.e. it has a smaller standard error), but that is is more biased (about 9.9% of the SE).

The 95% confidence interval for the trimmed mean is (13.67, 23.22). We are 95% confident that the trimmed mean chlorine concentration is between 13.67 and 23.22.

Exercise 12

Part a.

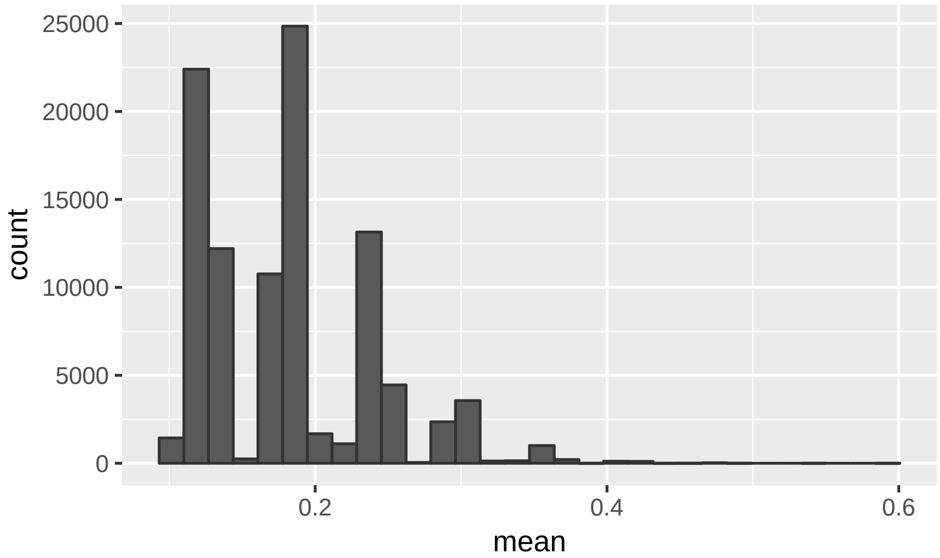
```
ggplot(data = FishMercury, mapping = aes(x = Mercury)) +
  geom_histogram(colour = "gray20", binwidth = 0.05)
```



The most obvious feature of this distribution is the outlier at 1.87. The outlier makes it hard to interpret the histogram, but the rest of the distribution appears to be unimodal.

Part b.

```
N <- 10^5
boot5.12b <- numeric(N)
for(i in 1:N)
{
  x <- sample(FishMercury$Mercury, replace = TRUE)
  boot5.12b[i] <- mean(x)
}

ggplot(data = data.frame(mean = boot5.12b), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20")
```



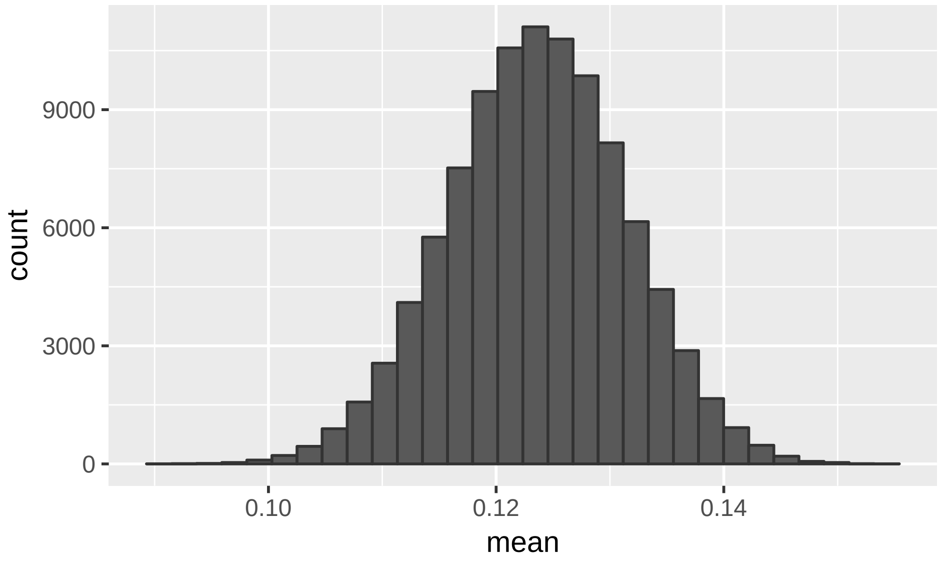
```
# bias
bias5.12b <- mean(boot5.12b) - mean(boot5.12b)
bias5.12b

## [1] 0
```

```
# SE
se5.12b <- sd(boot5.12b)
se5.12b

## [1] 0.05751882

# bias/SE
bias5.12b/se5.12b

## [1] 0

# CI
ci5.12b <- quantile(boot5.12b, probs = c(0.025, 0.975))
ci5.12b

##      2.5%     97.5%
## 0.1121333 0.3059667
```

The bootstrap distribution appears to be multimodal and right skewed when we bootstrap with the outlier. The standard error is 0.0575, and we find the 95% confidence interval to be (0.112, 0.306).

Part c.

```
N <- 10^5
boot5.12c <- numeric(N)
mercury_no_outlier <- subset(FishMercury, select = Mercury, subset = Mercury < 1.87, drop = TRUE)
for(i in 1:N)
{
  x <- sample(mercury_no_outlier, replace = TRUE)
  boot5.12c[i] <- mean(x)
}

ggplot(data = data.frame(mean = boot5.12c), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20")
```



```
# bias
bias5.12c <- mean(boot5.12c) - mean(boot5.12c)
bias5.12c

## [1] 0
```

```
# SE
se5.12c <- sd(boot5.12c)
se5.12c

## [1] 0.007810405

# bias/SE
bias5.12c/se5.12c

## [1] 0

# CI
ci5.12c <- quantile(boot5.12c, probs = c(0.025, 0.975))
ci5.12c

##      2.5%     97.5%
## 0.1081724 0.1387586
```

After removing the outlier, we now obtain a unimodal and symmetric distribution. The standard error is 0.0078, and we find the 95% confidence interval to be (0.108, 0.139).

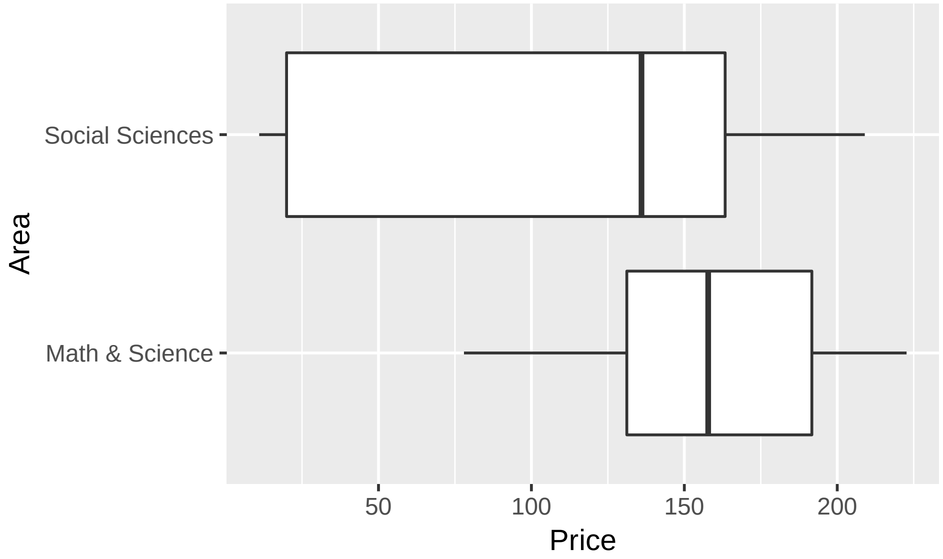
Part d.

The outlier was highly influential on the results. We saw that it impacted the shape of the distribution and inflated the standard error, which lead to a wide confidence interval when the outlier was included.

Exercise 17

Part a.

```
ggplot(data = BookPrices, mapping = aes(x = Area, y = Price)) +
  geom_boxplot() +
  coord_flip()
```



```
library(dplyr)
BookPrices %>%
  group_by(Area) %>%
  summarise(mean = mean(Price), sd = sd(Price))
```

```
## # A tibble: 2 × 3
##               Area     mean       sd
##             <fctr>    <dbl>    <dbl>
## 1  Math & Science 156.7341 39.14483
## 2 Social Sciences  98.9900 71.91385
```

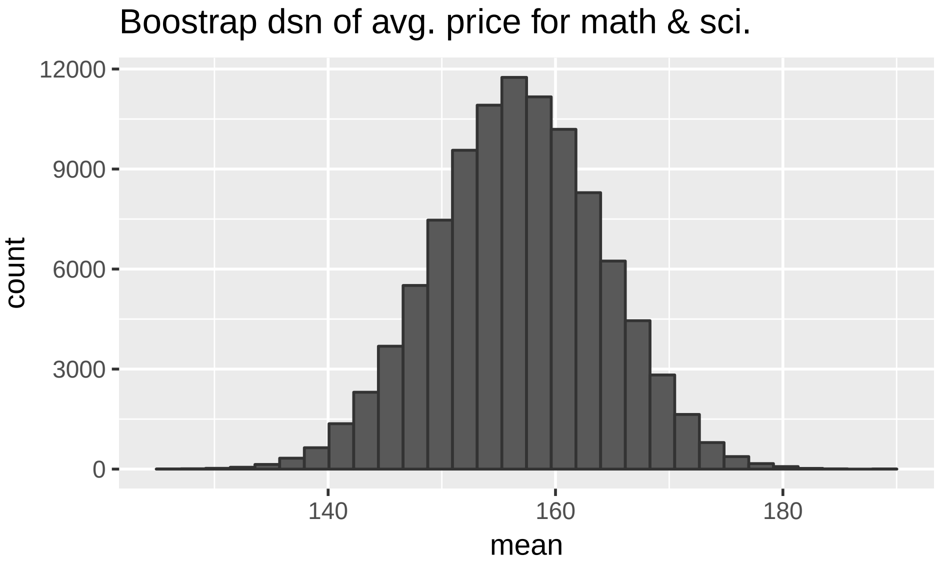
We find that the average price of a textbook in mathematics and the sciences is $156.73 while it is $98.99 for the social sciences. Further, the standard deviation of the prices is $39.14 and $71.91 for mathematics and the sciences, and the social sciences, respectively.

From the side-by-side boxplots, we see that

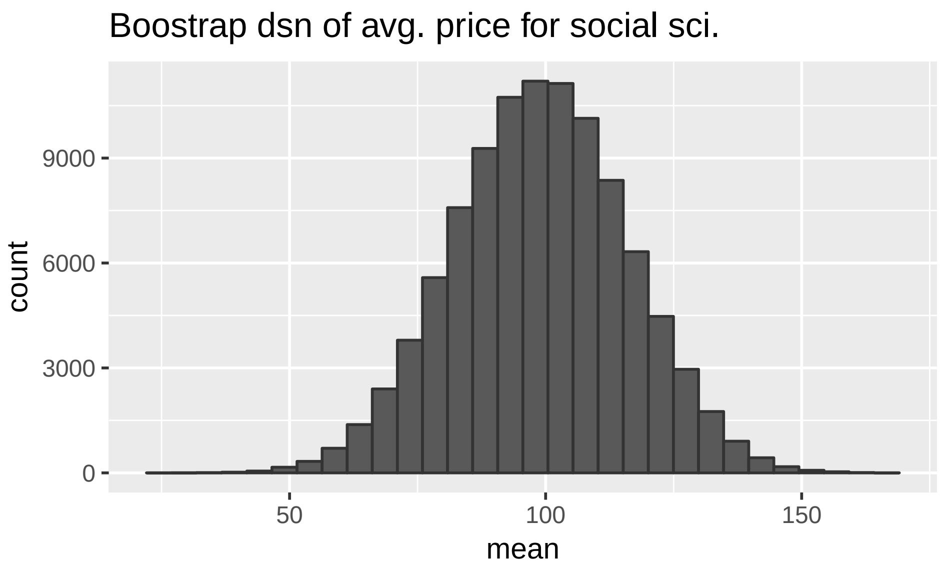
- The distribution of prices for the social sciences appears to be skewed to the left.
- The distribution of prices for mathematics and the sciences appears to be roughly symmetric.
- There are many (>25%) social science textbooks for less than $50, while there are no textbooks for mathematics and the sciences that cost less than $50.

Part b.

```
N <- 10^5
math <- subset(BookPrices, select = Price, subset = Area == "Math & Science", drop = TRUE)
socsci <- subset(BookPrices, select = Price, subset = Area == "Social Sciences", drop = TRUE)
boot_math   <- numeric(N)
boot_socsci <- numeric(N)
for(i in 1:N)
{
  boot_math[i] <- mean(sample(math, replace = TRUE))
  boot_socsci[i] <- mean(sample(socsci, replace = TRUE))
}

ggplot(data = data.frame(mean = boot_math), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20") +
  ggtitle("Boostrap dsn of avg. price for math & sci.")
```



```
ggplot(data = data.frame(mean = boot_socsci), mapping = aes(x = mean)) +
  geom_histogram(colour = "gray20") +
  ggtitle("Boostrap dsn of avg. price for social sci.")
```

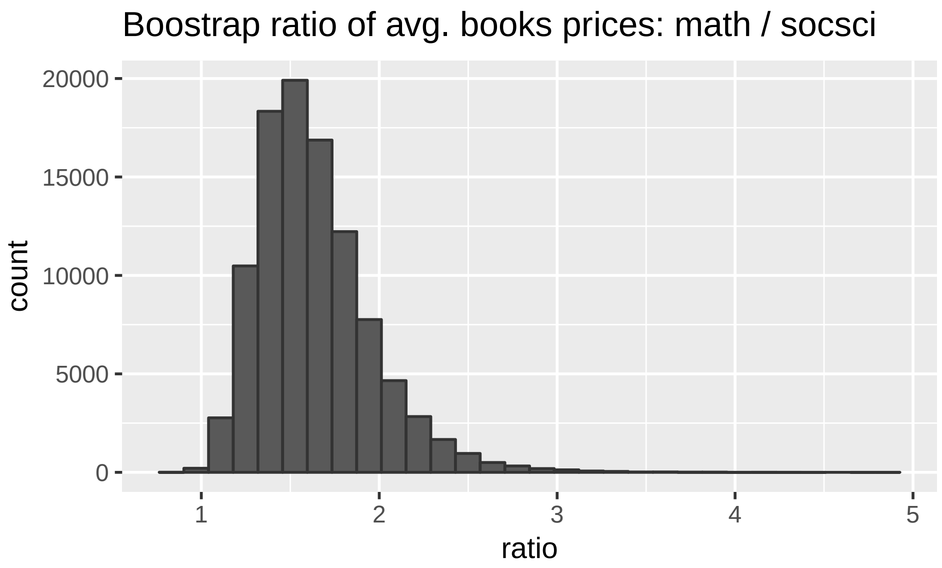


```
# bias
bias_math <- mean(boot_math) - mean(math); bias_math

## [1] 0.01809417

bias_socsci <- mean(boot_socsci) - mean(socsci); bias_socsci

## [1] -0.005500518

# SE
se_math <- sd(boot_math); se_math

## [1] 7.394703

se_socsci <- sd(boot_socsci); se_socsci

## [1] 16.96735

# bias/SE
bias_math / se_math

## [1] 0.00244691

bias_socsci / se_socsci

## [1] -0.0003241825
```

We see that both bootstrap distributions of the means are unimodal and symmetric, and that the bias represents a very small fraction of the standard error in each case.

Part c.

```
N <- 10^5
boot_ratio   <- numeric(N)
for(i in 1:N)
{
  boot_ratio[i] <- mean(sample(math, replace = TRUE)) / mean(sample(socsci, replace = TRUE))
}

ggplot(data = data.frame(ratio = boot_ratio), mapping = aes(x = ratio)) +
  geom_histogram(colour = "gray20") +
  ggtitle("Boostrap ratio of avg. books prices: math / socsci")
```



```r
# bias
bias_ratio <- mean(boot_ratio) - mean(math) / mean(socsci); bias_ratio

## [1] 0.05160187

# SE
se_ratio <- sd(boot_ratio); se_ratio

## [1] 0.3212451

# bias/SE
bias_ratio / se_ratio

## [1] 0.1606309
```

We see that the bootstrap distribution of the ratio of means is unimodal and skewed to the right.

Part d.

```r
ci5.17d <- quantile(boot_ratio, probs = c(0.025, 0.975)); ci5.17d

##     2.5%    97.5% 
## 1.166876 2.404053
```

We are 95% confident that book prices for math/science are between 1.17 and 2.4 times higher than for the social sciences.

Part e.

From the above R code, we see that the bootstrap estimate of bias is 0.052, which represents about 16.1% of the standard error.