MATH 392 Problem Set 3

EJ Arce

12 February 2018

4.8

$$n = 20, \mu = 6, \sigma^2 = 10$$

$$P(\bar{X} \leq 4.6) = P(\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \leq \frac{4.6 - \mu}{\sigma/\sqrt{n}})$$

$$P(\bar{X} \leq 4.6) = P(\frac{\bar{X} - 6}{\sqrt{10}/\sqrt{20}} \leq \frac{4.6 - 6}{\sqrt{10}/\sqrt{20}})$$

```r
# Calculate Z score of interested statistic
z.obs <- (4.6-6)/(sqrt(10)/sqrt(20))
z.obs
```

```
## [1] -1.979899
```

$$P(\bar{X} \leq 4.6) = P(Z \leq -1.98)$$

```r
# Calculate probability using the cdf of N(0,1)
pnorm(z.obs, 0, 1)
```

```
## [1] 0.02385744
```

$$P(\bar{X} \leq 4.6) = .02385$$

4.9

$$f_X(x) = \frac{3}{16}(x-4)^2 for 2 \leq 6$$

Find E[X]:

$$E[X] = \int_2^6 x\frac{3}{16}(x-4)^2 dx$$

$$E[X] = \int_2^6 \frac{3}{16}x(x^2 - 8x + 16)dx$$

$$E[X] = \int_2^6 \frac{3}{16}x^3 - \frac{3}{2}x^2 + 3xdx$$

$$E[X] = \frac{3}{64}x^4 - \frac{1}{2}x^3 + \frac{3}{2}x^2|_2^6$$

$$E[X] = 4$$

Find V[X]:

$$V[X] = E[X^2] - E[X]^2$$

We already calculated that $E[X] = 4$, so $E[X]^2 = 16$. Now solve for $E[X^2]$:

$$E[X^2] = \int_2^6 x^2 f(x) dx$$

$$E[X^2] = \int_2^6 x^2 \frac{3}{16}(x-4)^2 dx$$

$$E[X^2] = \int_2^6 \frac{3}{16}x^4 - \frac{3}{2}x^3 + 3x^2 dx$$

$$E[X^2] = \frac{3}{80}x^5 - \frac{3}{8}x^4 + x^3|_2^6$$

```
# Calculate
e.xsq <- (3*(6^5)/80 - 3*(6^4)/8 +6^3) -
  (3*(2^5)/80 - 3*(2^4)/8 +(2^3))
e.xsq
```

```
## [1] 18.4
```

```
sq.ex <- 4^2
var.x <- e.xsq-sq.ex
sd.x <- sqrt(var.x)
sd.x
```

```
## [1] 1.549193
```

Thus $V[X] = 2.4$, so SD[X] $= \sqrt{2.4} = 1.549$. Now, $n = 244, \mu = 4, \sigma^2 = 2.4$.

$$P(\bar{X} \geq 4.2) = P(\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \geq \frac{4.2 - \mu}{\sigma/\sqrt{n}})$$

$$P(\bar{X} \geq 4.2) = P(\frac{\bar{X} - 4}{\sqrt{2.4}/\sqrt{244}} \geq \frac{4.2 - 4}{\sqrt{2.4}/\sqrt{244}})$$

$$P(\bar{X} \geq 4.2) = P(Z \geq \frac{4.2 - 4}{\sqrt{2.4}/\sqrt{244}})$$

```
# Calculate Z score of interested statistic
z.obs <- (4.2-4)/(sqrt(2.4)/sqrt(244))
z.obs
```

```
## [1] 2.016598
```

```
# Calculate probability using cdf from N(0,1)
1 - pnorm(z.obs,0,1)
```

```
## [1] 0.02186875
```

4.12

a

Let X be a random sample of size 30 from the exponential distribution with rate $\lambda = .1$. The expected value of the sample mean is the same as the expected value of the population, so E[X] $= \frac{1}{\lambda} = 10$

b

```r
# Run simulation
nsim <- 1000
n <- 30
rate <- 1/10
means <- rep(NA,nsim)
for(i in 1:nsim){
  sample <- rexp(n,rate)
  means[i] <- mean(sample)
}
sum(means >= 12)/nsim
```

```
## [1] 0.121
```

c

Since 12.1% of the samples had means of 12 or greater, this observation is not that unusual.

4.13

a

Since X ~ $N(20, 8^2)$ and Y ~ $N(16, 7^2)$ are independent variables, and W $= \bar{X} + \bar{Y}$, then W ~ $N(36, \frac{8^2}{10} + \frac{7^2}{15})$

b

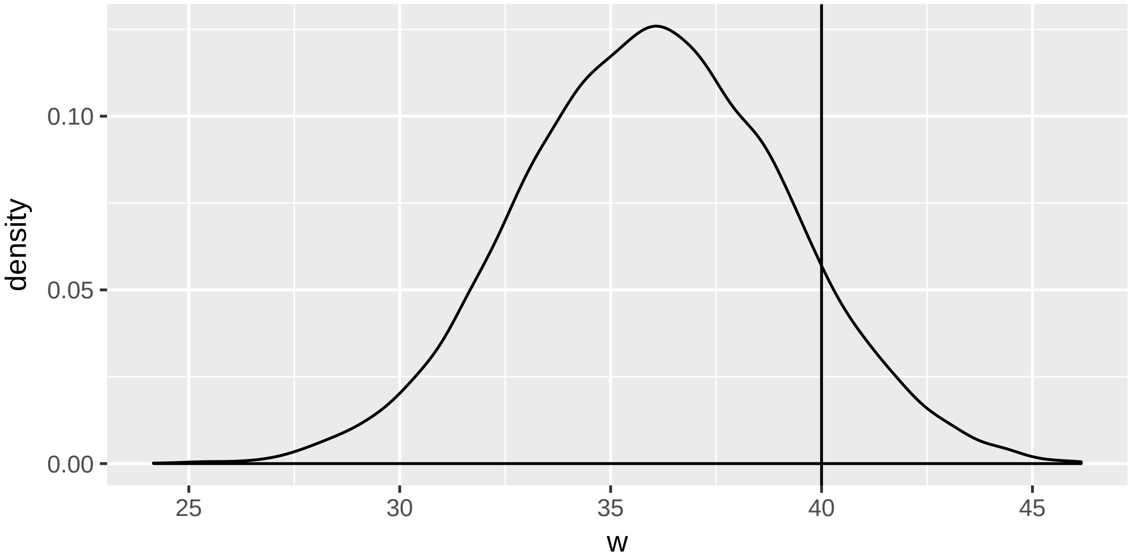
```r
# Run simulation of sampling distribution
nsim <- 10000
w <- rep(NA,nsim)
for(i in 1:nsim){
  x <- rnorm(10,20,8)
  y <- rnorm(15,16,7)
  w[i] <- mean(x) + mean(y)
}
# Compute mean and standard error
mean(w) # Similar to theoretical mean of 36
```

```
## [1] 36.01261
```

```r
sd(w) # Similar to theoretical standard error of 3.109
```

```
## [1] 3.132671
```

```r
# Plot sampling distribution
w <- data.frame(w)
ggplot(w,aes(x=w)) +
  geom_density() +
  geom_vline(xintercept=40)
```



c

```r
# Find simulated probability
(sum(w<40) + 1) /( nsim + 1)
```

```
## [1] 0.8975102
```

```r
# Calculate exact answer
se <- sqrt(((8^2)/10) +((7^2)/15))
pnorm(40,36,se)
```

```
## [1] 0.9008718
```

The two values are not too far off.

4.18

a

```r
# Simulate sampling distribution
nsim <- 10000
n <- 30
rate <- 1/3
x.bars <- rep(NA,nsim)
for(i in 1:nsim){
  sample <- rexp(n,rate)
  x.bars[i] <- mean(sample)
}
```

b

```r
# Compute and compare simulated mean and standard error with theoretical results
mean.sim <- mean(x.bars)
se.sim <- sd(x.bars)
mean.theory <- 1/rate
se.theory <- (1/rate)/sqrt(n)
mean.sim
```

```
## [1] 2.996693
```

```r
mean.theory
```

```
## [1] 3
```

```r
se.sim
```

```
## [1] 0.5416903
```

```r
se.theory
```

```
## [1] 0.5477226
```

c

```r
# Calculate simulated probability
d <- data.frame(x.bars)
(sum(x.bars <= 3.5) + 1) / (nsim + 1)
```

```
## [1] 0.8244176
```

d

$$n = 30, X\ exp(1/3), \mu = 1/3, \sigma^2 = 9$$

$$P(\bar{X} \leq 3.5) = P(\frac{\bar{X} - \mu}{\sqrt{\sigma^2}/\sqrt{n}} \leq \frac{3.5 - \mu}{\sqrt{\sigma^2}/\sqrt{n}}) = P(\frac{\bar{X} - 3}{\sqrt{9}/\sqrt{30}} \leq \frac{3.5 - 3}{\sqrt{9}/\sqrt{30}})$$

```r
# Calculate z score we are testing
z.test <- (3.5-3)/(sqrt(9)/sqrt(30))
z.test
```

```
## [1] 0.9128709
```

$$P(\bar{X} \leq 3.5) = P(Z \leq .9129)$$

```r
# Calculate approximated probability
pnorm(z.test,0,1)
```

```
## [1] 0.8193448
```

$P(\bar{X} \leq 3.5) = .8193$. The approximated result is similar to the simulated probability of .8252.

4.20

$LetX_1, ..., X_n$ be continuouus and i.i.d. random variables with pdf f and cdf F. Show that the pdf's for X_{min} and X_{max} are

$$f_{min}(x) = n(1 - F(x))^{n-1} f(x)$$

$$f_{max}(x) = nF(x)^{n-1}(x) f(x)$$

First, show the pdf of X_{max}:

$$F_{max}(x) = P(max(X_1, ..., X_n) \leq x)$$

$$F_{max}(x) = P(X_1 \leq x, ..., X_n \leq x)$$

Because the variables are i.i.d.,

$$F_{max}(x) = P(X_1 \leq x)...P(X_n \leq x)$$

$$F_{max}(x) = F(x)...F(x) = F(x)^n$$

Now, differentiate to find $f_{max}(x)$:

$$f_{max}(x) = \frac{\partial}{\partial x} F_{max}(x) = nF(x)^{n-1} f(x)$$

Now show the pdf$f_{min}(x)$:

$$F_{min}(x) = P(min(X_1, ..., X_n) \leq x)$$

At least one of the variables $X_i \leq x$, so we can use the probability that none of the random variables will be less than x, and subtract that from 1:

$$F_{min}(x) = (1 - P(X_1 \leq x))...(1 - P(X_n \leq x))$$

$$F_{min}(x) = (1 - P(X \leq x))^n = (1 - F(x))^n$$

Now differentiate to find $f_{min}(x)$:

$$f_{min}(x) = \frac{\partial}{\partial x} F_{min}(x) = n(1 - F(x))^{n-1} f(x)$$

4.21

a

By theorem 4.1, $f_{max}(x) = nF(x)^{n-1}f(x)$. In this case, n = 2, and $f(x) = 2/x^2 for 1 \leq x \leq 2$. So

$$f_{max}(x) = 2(2/x^2)\int_1^x 2x^{-2}dx$$

$$f_{max}(x) = 2(2/x^2)(-2x^{-1}|_1^x)$$

$$f_{max}(x) = 2(2/x^2)(2(1-1/x))$$

$$f_{max}(x) = \frac{8-8/x}{x^2}$$

b

Solve for E[X]:

$$E[X] = \int_1^2 x f_{max}(x)dx$$

$$E[X] = \int_1^2 x\frac{8-8/x}{x^2}dx$$

$$E[X] = \int_1^2 8x^{-1} - 8x^{-2}dx$$

$$E[X] = 8lnx + 8x^{-1}|_1^2 = (8ln2 + 4) - (8ln1 + 8)$$

$$E[X] = (8ln2 + 4) - (8ln1 + 8) = 8ln2 + 4 - (0 + 8)$$

$$E[X] \approx 1.545.$$

5.2

```r
# Run simulation for a and b
dist <- c(1,3,4,6)
nsim <- 10000
means <- rep(NA,nsim)
maxs <- rep(NA,nsim)
for(i in 1:nsim){
  boot <- sample(dist,4,replace=T)
  means[i] <- mean(boot)
  maxs[i] <- max(boot)
}
```

a

```
(sum(means == 1) + 1)/(nsim + 1)
```

```
## [1] 0.00269973
```

b

```
(sum(maxs == 6) + 1)/(nsim + 1)
```

```
## [1] 0.6867313
```

c

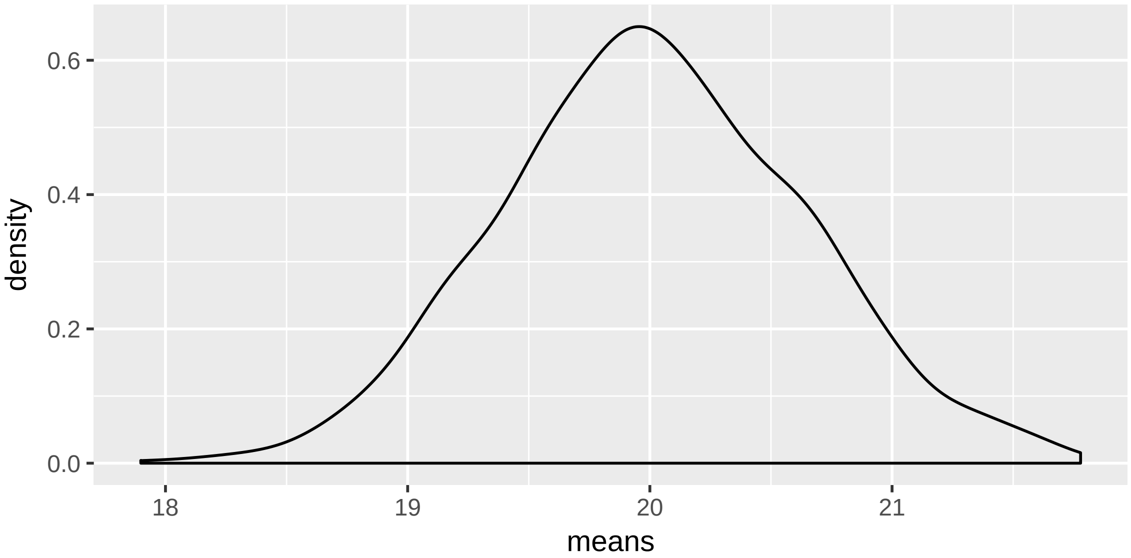
```
indicator <- rep(NA,nsim)
for(i in 1:nsim){
  boot <- sample(dist,4,replace=T)
  sum <- sum(boot < 2)
  if(sum == 2){
    indicator[i] <- 1
  }
  else{
    indicator[i] <- 0
  }
}
mean(indicator)
```

```
## [1] 0.2097
```

5.8

a

```
# Similate sampling distribution
nsim <- 1000
n <- 200
shape <- 5
rate <- 1/4
means <- rep(NA,nsim)
for(i in 1:nsim){
  sample <- rgamma(n,shape,rate)
  means[i] <- mean(sample)
}
df.sample <- data.frame(means)
samp.dist <- ggplot(df.sample, aes(x=means)) +
  geom_density()
samp.dist
```

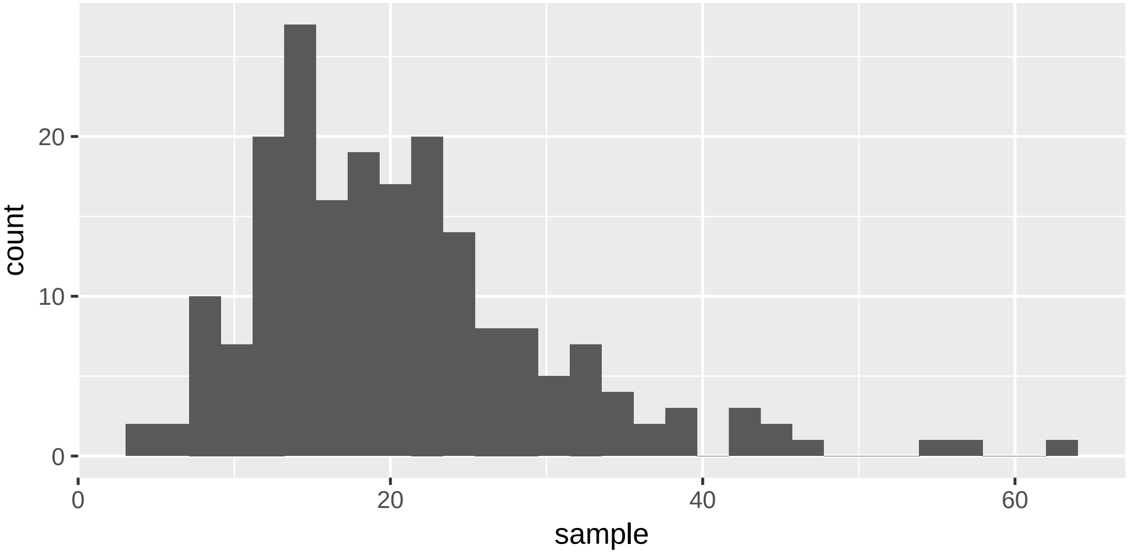


As expected, the distribution is approximately normal and is centered at 20.

b

```
sample <- rgamma(n,shape,rate)
df <- data.frame(sample)
ggplot(df,aes(x=sample)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mean(sample)
```

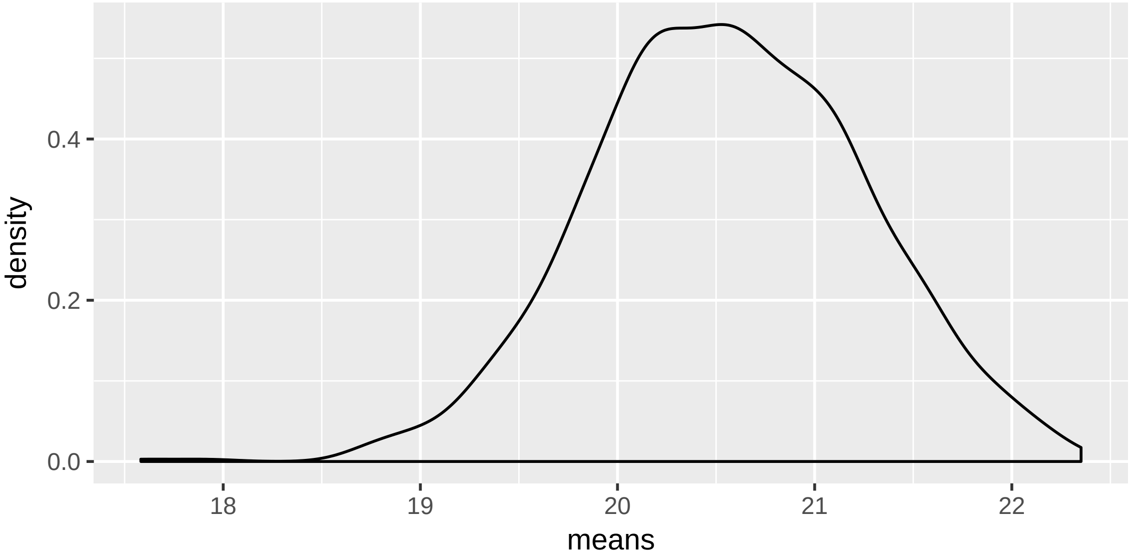
```
## [1] 20.55187
```

```
sd(sample)
```

```
## [1] 9.648377
```

c

```
means <- rep(NA,nsim)
for(i in 1:nsim){
  boot <- sample(sample,n,replace=T)
  means[i] <- mean(boot)
}
df.boot <- data.frame(means)
boot.dist <- ggplot(df.boot,aes(x=means)) +
  geom_density()
mean <- mean(df.boot$means)
se <- sd(df.boot$means)
boot.dist
```



```
mean
```

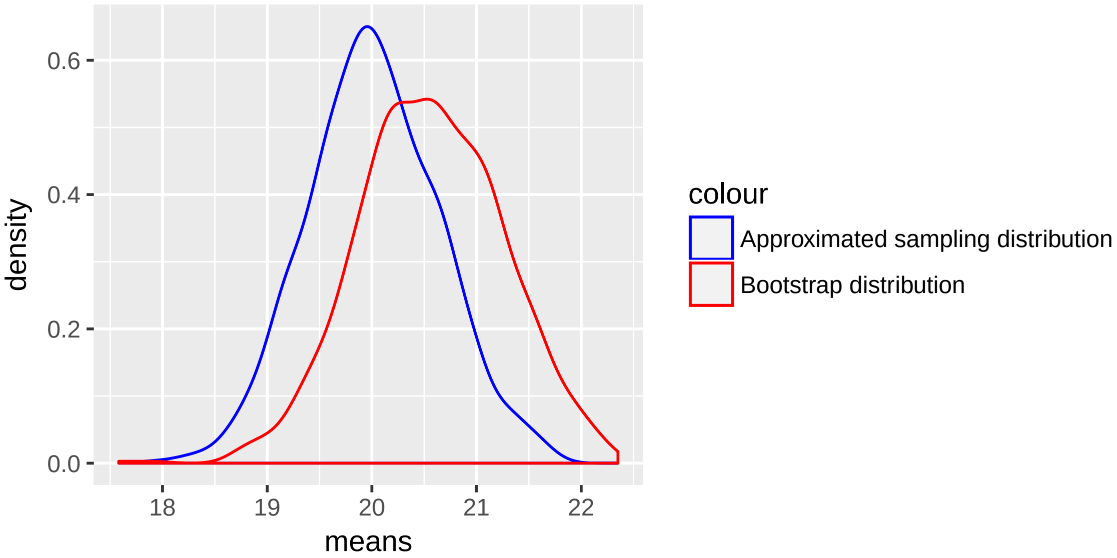
```
## [1] 20.55716
```

```
se
```

```
## [1] 0.6874307
```

d

```
ggplot(df.sample, aes(x=means, col ="blue")) +
  geom_density() +
  geom_density(data = df.boot, aes(col="red")) +
  scale_color_manual(labels = c("Approximated sampling distribution",
                                "Bootstrap distribution"),
                     values = c("blue","red"))
```

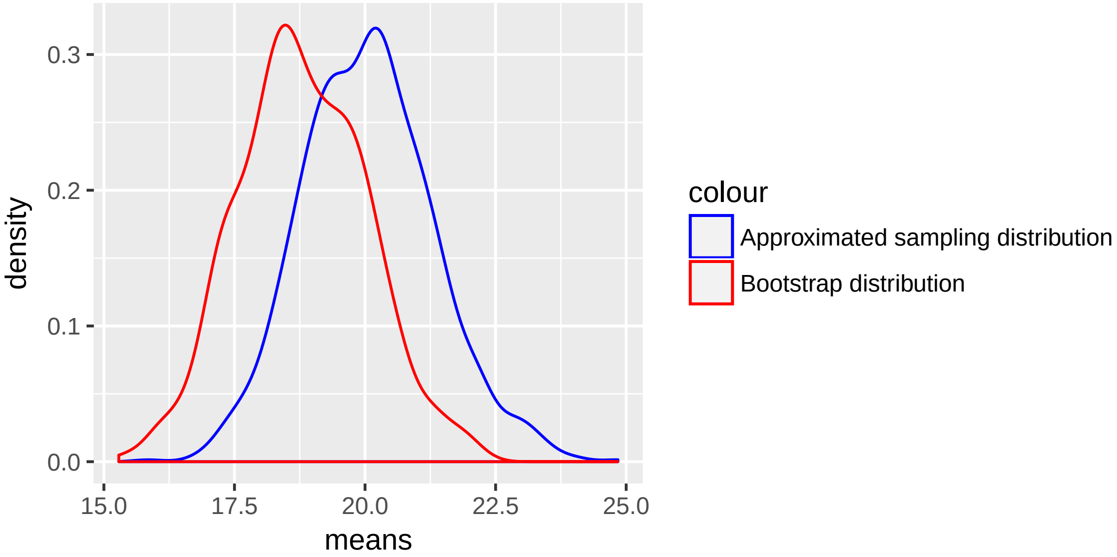


The two distributions appear similar in skew and shape, but the bootstrap distribution estimates the mean of the population to be higher than the sampling distribution's estimate.

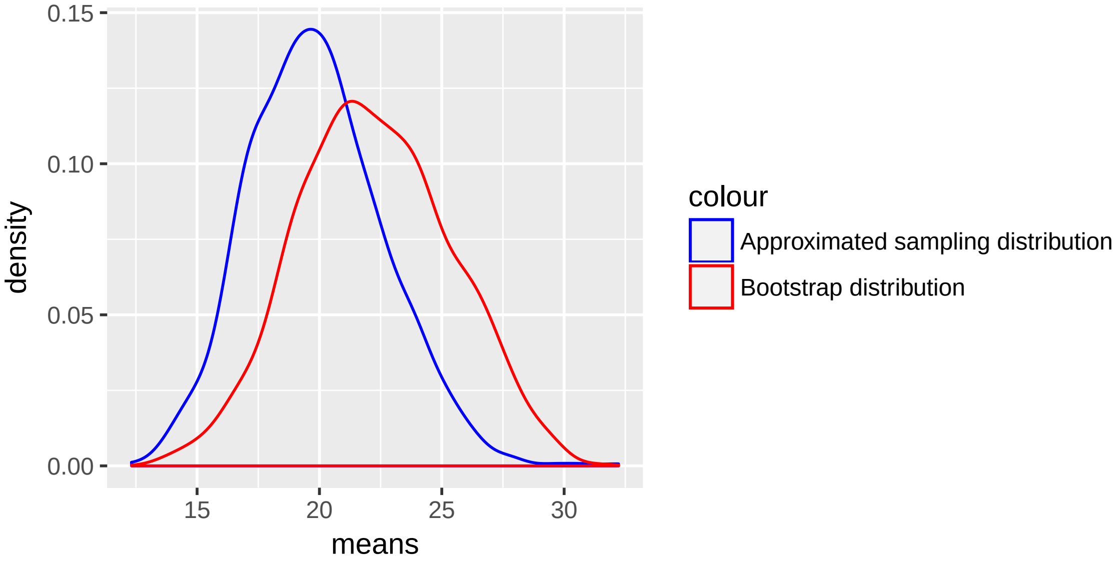
e

The codes producing the graphs below are exactly the same as the codes for problems a-d, exect n is changed to 50 and 10, respectively.

n = 50



n = 10

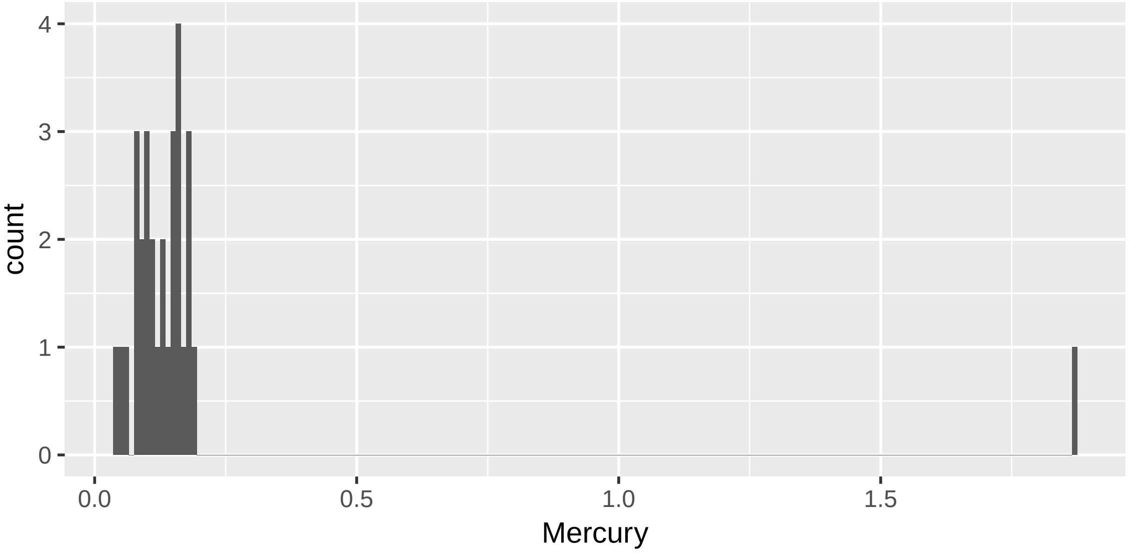


For both the sampling distribution and the bootstrap distribution, the variance of the mean estimates increases as the sample size, n, decreases. Changing the sample size does not appear to change the mean estimate in any particular way. In each case, the sampling distribution provides a better estimate of the mean than the bootstrap distribution.

5.12

a

```
ggplot(FishMercury,aes(x=Mercury)) +
  geom_histogram(binwidth=.01)
```



The data includes 29 observations with a mercury level of less than .25 parts per million and 1 observation with a mercury level of 1.87 parts per million.

b

```r
data(FishMercury)
nsim <- 10000
n <- 30
means <- rep(NA,nsim)
for(i in 1:nsim){
  boot <- sample(FishMercury$Mercury, n, replace=T)
  means[i] <- mean(boot)
}
se.boot <- sd(means)
se.boot
```

```
## [1] 0.0570548
```

```r
CI <- quantile(means, probs = c(.025,.975))
CI
```

```
##      2.5%     97.5%
## 0.1121992 0.3053025
```

c

```r
# Remove outlier
FishMercury2 <- FishMercury %>%
  filter(Mercury < 1)

# Repeat simulation
nsim <- 10000
n <- 30
means <- rep(NA,nsim)
for(i in 1:nsim){
  boot <- sample(FishMercury2$Mercury, n, replace=T)
  means[i] <- mean(boot)
}
se.boot <- sd(means)
se.boot
```

```
## [1] 0.007688291
```

```r
CI <- quantile(means, probs = c(.025,.975))
CI
```

```
##      2.5%     97.5%
## 0.1087000 0.1386675
```

d

The standard error greatly reduced, now that the outlier can't affect the mean estimate in each bootstrap sample.