

# Chapter 3 Confidence Intervals via Bootstrapping

```
suppressPackageStartupMessages({  
  library(ggplot2)    # graphing functions  
  library(dplyr)      # data summary tools  
  library(boot)       # bootstrap  
  library(car)        # for the bootstrap confint() function  
})  
  
# Set default behavior of ggplot2 graphs to be black/white theme  
theme_set(theme_bw())
```

## 3.1 Theory of Bootstrapping

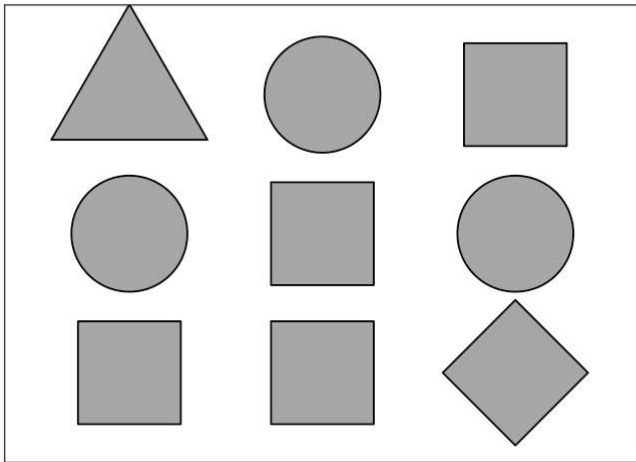
Suppose that we had a population of interest and we wish to estimate the mean of that population (the population mean we'll denote as  $\mu$ ). We can't observe every member of the population (which would be prohibitively expensive) so instead we take a random sample and from that sample calculate a sample mean (which we'll denote  $\bar{x}$ ). We believe that  $\bar{x}$  will be a good estimator of  $\mu$ , but it will vary from sample to sample and won't be exactly equal to  $\mu$ .

Next suppose we wish to ask if a particular value for  $\mu$ , say  $\mu_0$ , is consistent with our observed data? We know that  $\bar{x}$  will vary from sample to sample, but we have no idea how much it will vary between samples. However, if we could understand how much  $\bar{x}$  varied sample to sample, we could answer the question. For example, suppose that  $\bar{x} = 5$  and we know that  $\bar{x}$  varied about  $\pm 2$  from sample to sample. Then I'd say that possible values of  $\mu_0$  in the interval 3 to 7 ( $5 \pm 2$ ) are reasonable values for  $\mu$  and anything outside that interval is not reasonable.

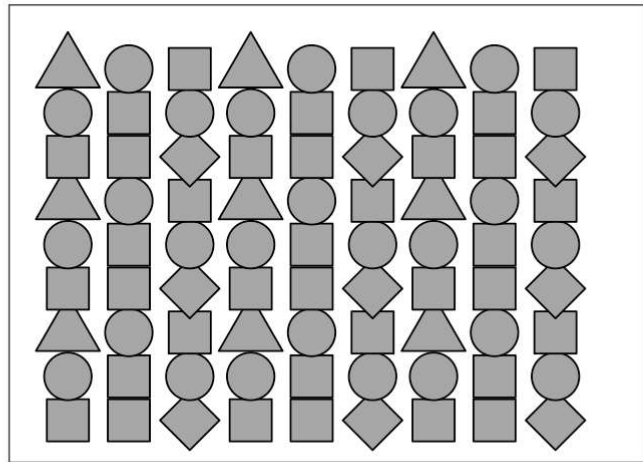
Therefore, if we could take many, many repeated samples from the population and calculate our test statistic  $\bar{x}$  for each sample, we could rule out possible values of  $\mu$ . Unfortunately we don't have the time or money to repeatedly sample from the actual population, but we could sample from our best approximation to what the population is like.

Suppose we were to sample from a population of shapes, and we observed 4/9 of the sample were squares, 3/9 were circles, and a triangle and a diamond. Then our best guess of what the population that we sampled from was a population with 4/9 squares, 3/9 circles, and 1/9 of triangles and diamonds.

Sample



Approximate Population



Using this approximated population (which is just many many copies of our sample data), we can repeatedly sample  $\bar{x}^*$  values to create an estimate of the sampling distribution of  $\bar{x}$ .

Because our approximate population is just an infinite number of copies of our sample data, then sampling from the approximate population is equivalent to sampling with replacement from our sample data. If I take  $n$  samples from  $n$  distinct objects with replacement, then the process can be thought of as mixing the  $n$  objects in a bowl and taking an object at random, noting which it is, replace it into the bowl, and then draw the next sample. Practically, this means some objects will be selected more than once and some will not be chosen at all. To sample our observed data with replacement, we can use the `sample()` function in R.

```
names=c('Alison','Brandon','Casey','Derek','Elise')
sample(names, length(names), replace=T)
```

```
## [1] "Elise" "Alison" "Elise" "Alison" "Derek"
```

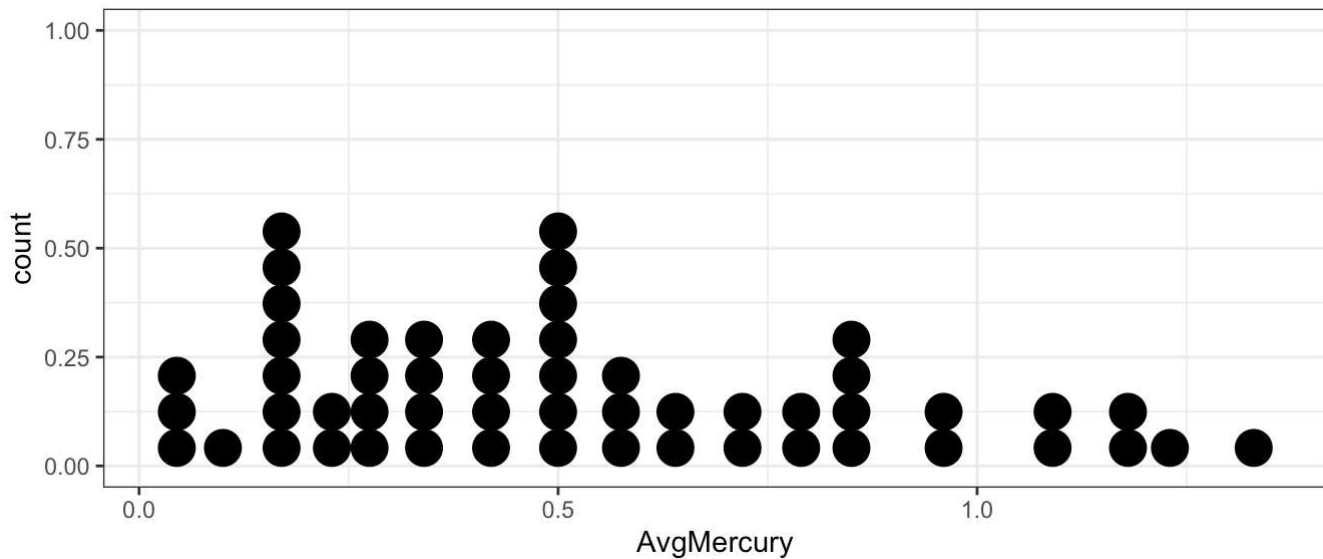
Notice Alison has selected twice, while Brandon has not been selected at all.

The sampling from the estimated population via sampling from the observed data is called **bootstrapping** because we are making no distributional assumptions about where the data came from, and the idiom “Pulling yourself up by your bootstraps” seemed appropriate.

### Example: Mercury Levels in Fish from Florida Lakes

A data set provided by the Lock<sup>5</sup> introductory statistics textbook looks at the mercury levels in fish harvested from lakes in Florida. There are approximately 7,700 lakes in Florida that are larger than 10 acres. As part of a study to assess the average mercury contamination in these lakes, a random sample of  $n = 53$  lakes, an unspecified number of fish were harvested and the average mercury level (in ppm) was calculated for fish in each lake. The goal of the study was to assess if the average mercury concentration was greater than the 1969 EPA “legally actionable level” of 0.5 ppm.

```
# read the Lakes data set
Lakes <- read.csv('http://www.lock5stat.com/datasets/FloridaLakes.csv')
# make a nice picture... dot plots are very similar to histograms
# dot plots can be informative for small samples. Unfortunately the
# y-axis is not meaningful in the geom_dotplot() version because of ... reasons.
ggplot(Lakes, aes(x=AvgMercury)) +
  geom_dotplot()
```



We can calculate mean average mercury level for the  $n = 53$  lakes

```
Lakes %>% summarise(xbar = mean( AvgMercury ))
```

```
##           xbar
```

```
## 1 0.5271698
```

The sample mean is greater than 0.5 but not by too much. Is a true population mean concentration  $\mu_{Hg}$  that is 0.5 or less incompatible with our observed data? Is our data sufficient evidence to conclude that the average mercury content is greater than 0.5? Perhaps the true average mercury content is less than (or equal to) 0.5 and we just happened to get a random sample that with a mean greater than 0.5?

## 3.2 Conducting a Bootstrap

The first step in answering these questions is to create an estimate of the sampling distribution of  $\bar{x}_{Hg}$ . To do this, we will sample from the approximate population of lakes, which is just many many replicated copies of our sample data. There are many ways to bootstrap using R, and chosen here is to introduce the package `boot` for conducting the bootstrap for us with minimal code. For alternative methods using base R or the package `mosaic`, see Appendix A.

```
library(boot)
```

To use the `boot()` function within the `boot` package, we will have to define a function for the resampling to occur. Below, we create the function `mean.function`, that accepts a vector (our observations) and calculates the mean. The index is so that `boot()` can do the resampling. How do you think we could change this to bootstrap different statistics?

```
# functions we defined to be passed into boot() need to take a whatever data
# you'll ultimately pass into boot() (here we call that data x) and set of
# indices.
mean.function <- function(x, index) {
  d <- x[index]      # This first line will go in ever bootstrap function you make.
  return(mean(d))
}
```

Once you have defined what you would like to bootstrap, the function `boot()` is a simple call in R, and produces the number of iterations  $R$  we choose. Let us try running  $R = 10000$  bootstrap iterations.

```
# create the Estimated Sampling Distribution of xbar
BootDist <- boot(data = Lakes$AvgMercury, statistic = mean.function, R=10000)
```

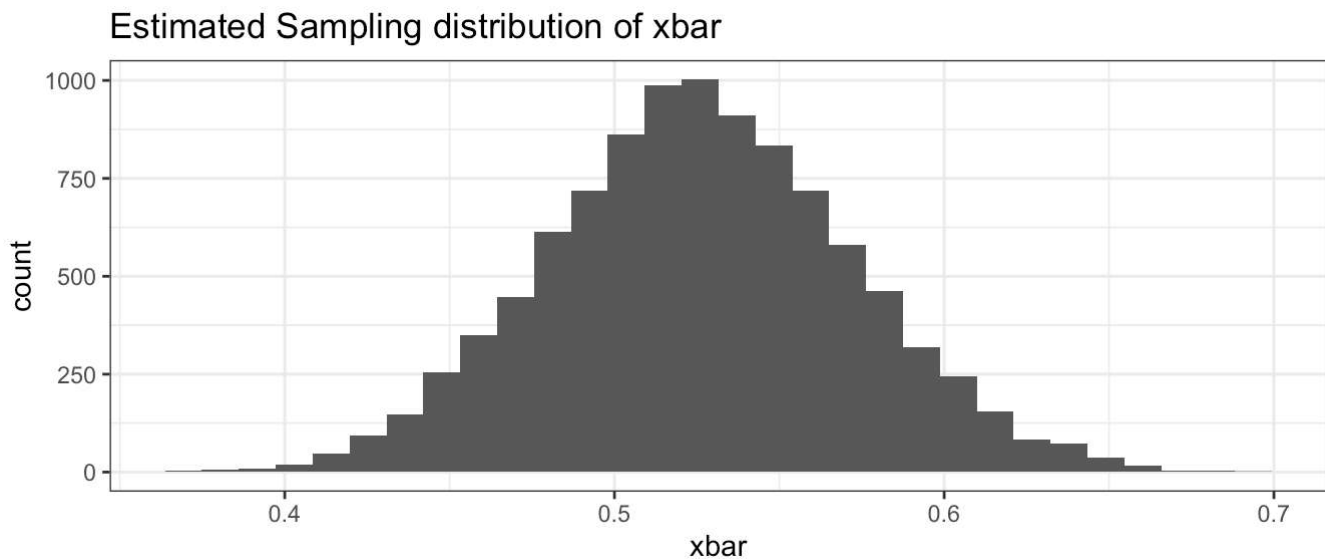
There are many outputs available within the output of `boot()`. We are interested in the calculated statistic for each redraw, which is saved within the output as the variable  $t$ . We can place the calculated means for each redraw into a data frame and produce a visualization of the estimated sampling distribution of  $\bar{x}$ .

```
# The first few calculated means.
head(BootDist$t)
```

```
##           [,1]
## [1,] 0.5209434
## [2,] 0.4884906
## [3,] 0.5766038
## [4,] 0.5275472
## [5,] 0.5830189
## [6,] 0.5245283
```

```
# show a histogram of the estimated sampling distribution of xbar
```

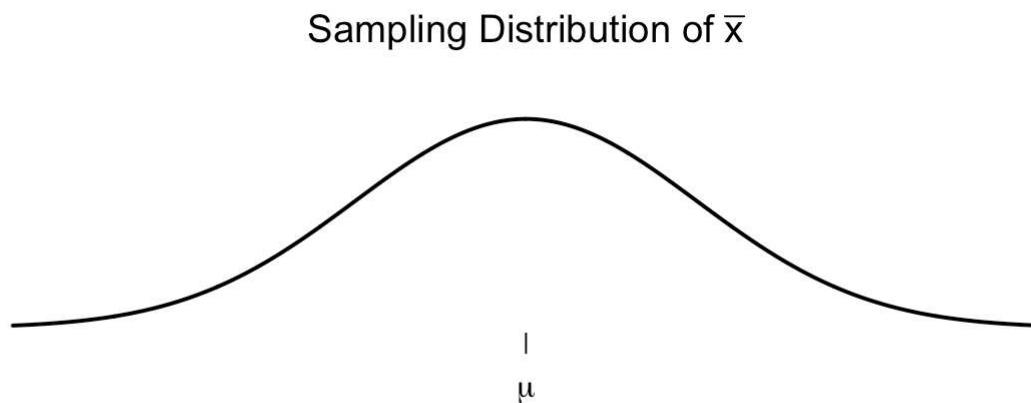
```
BootDist.graph <- data.frame(xbar=BootDist$t)
ggplot(BootDist.graph, aes(x=xbar)) +
  geom_histogram() +
  ggtitle('Estimated Sampling distribution of xbar' )
```



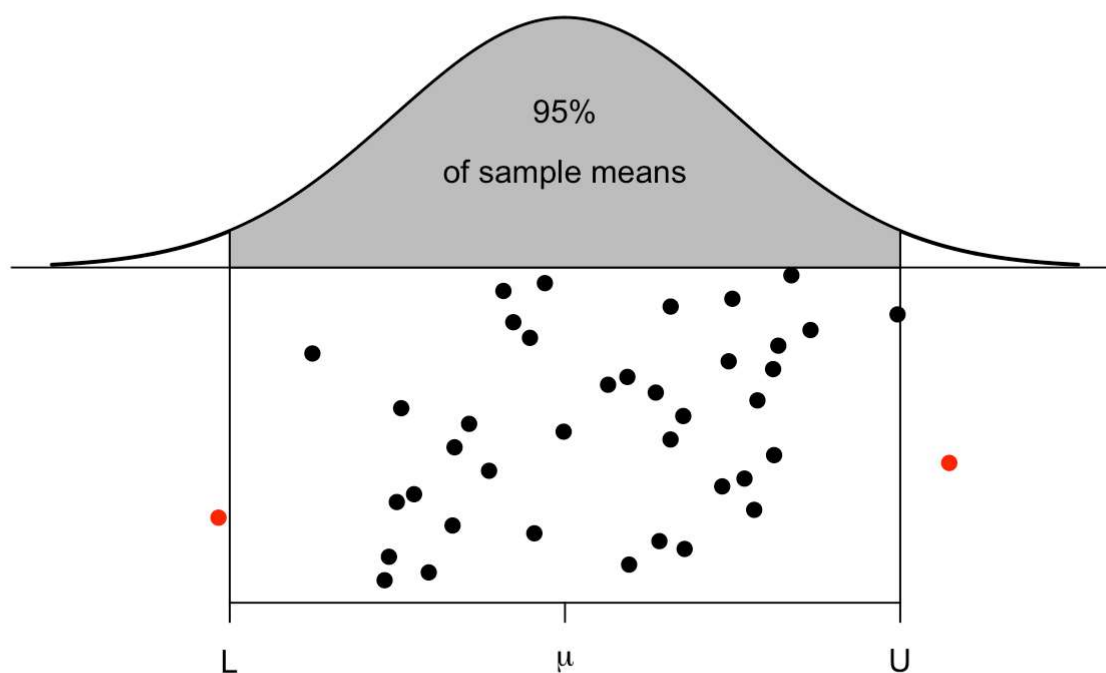
### 3.3 Quantile-based Confidence Intervals

In many cases we have seen, the sampling distribution of a statistic is centered on the parameter we are interested in estimating and is symmetric about that parameter. There are actually several ways to create a confidence interval from the estimated sampling distribution. The method presented here is called the “percentile” method and works when the sampling distribution is

symmetric and the estimator we are using is unbiased. For example, we expect that the sample mean  $\bar{x}$  should be a good estimate of the population mean  $\mu$  and the sampling distribution of  $\bar{x}$  should look something like the following.

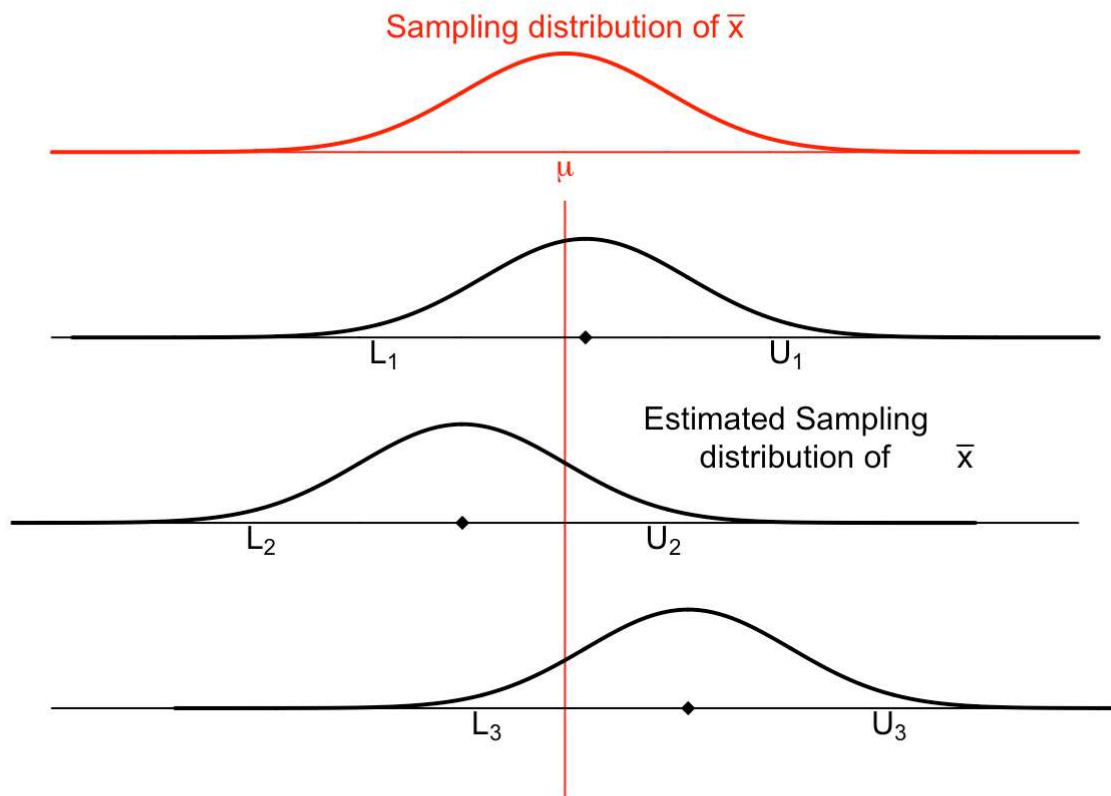


There are two points, (call them  $L$  and  $U$ ) where for our given sample size and population we are sampling from, where we expect that 95% of the sample means to fall within. That is to say,  $L$  and  $U$  capture the middle 95% of the sampling distribution of  $\bar{x}$ .

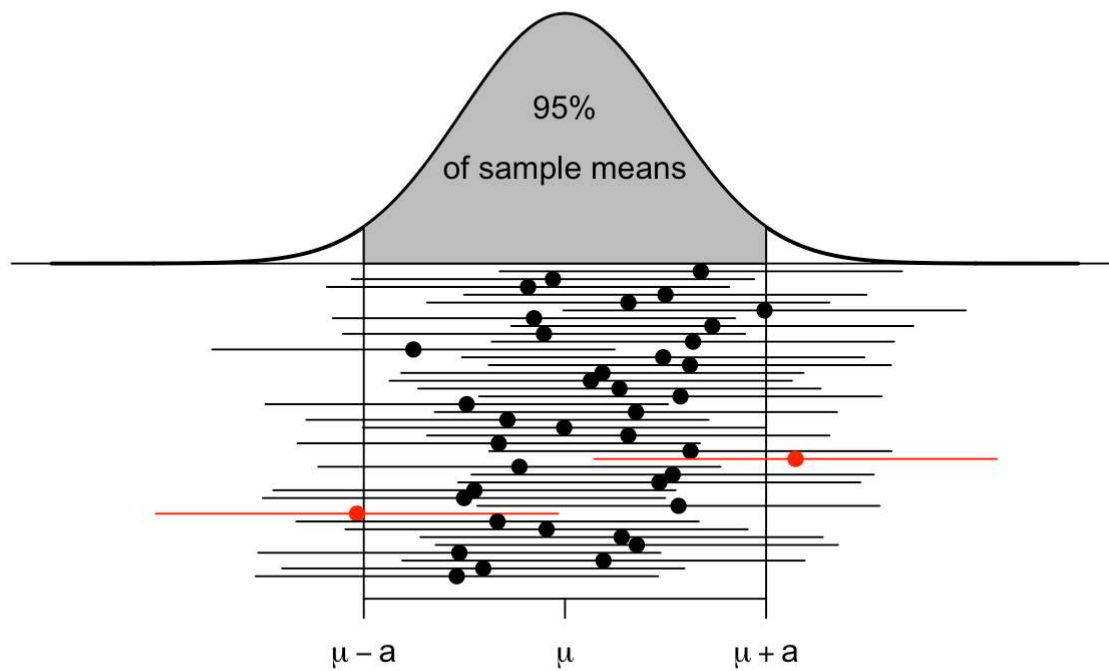
Sampling Distribution of  $\bar{x}$ 

These sample means are randomly distributed about the population mean  $\mu$ . Given our sample data and sample mean  $\bar{x}$ , we can examine how our simulated values of  $\bar{x}^*$  vary about  $\bar{x}$ . I expect that these simulated sample means  $\bar{x}^*$  should vary about  $\bar{x}$  in the same way that  $\bar{x}$  values vary around  $\mu$ . Below are three estimated sampling distributions that we might obtain from three different samples and their associated sample means.





For each possible sample, we could consider creating the estimated sampling distribution of  $\bar{X}$  and calculating the  $L$  and  $U$  values that capture the middle 95% of the estimated sampling distribution. Below are twenty samples, where we've calculated this interval for each sample.



Most of these intervals contain the true parameter  $\mu$ , that we are trying to estimate. In practice, I will only take one sample and therefore will only calculate one sample mean and one interval, but I want to recognize that the method I used to produce the interval (i.e. take a random sample, calculate the mean and then the interval) will result in intervals where only 95% of those intervals will contain the mean  $\mu$ . Therefore, I will refer to the interval as a 95% confidence interval.

After the sample is taken and the interval is calculated, the numbers lower and upper bounds of the confidence interval are fixed. Because  $\mu$  is a constant value and the confidence interval is fixed, nothing is changing. To distinguish between a future random event and the fixed (but unknown) outcome of if I ended up with an interval that contains  $\mu$  and we use the term confidence interval instead of probability interval.

```
# calculate the 95% confidence interval using middle 95% of xbars
# Calculate this by hand
quantile( BootDist$t, probs=c(.025, .975) )
```

```
##          2.5%      97.5%
## 0.4364104 0.6184953
```

```
# Or using the `car` package which also makes more robust intervals.
# By default it uses the 'bca' method which is generally more robust
# and should be preferred. We won't talk about the differences here, though.
confint( BootDist, level=.95, type='perc')
```

```
## Bootstrap percent confidence intervals
##
##          2.5 %      97.5 %
## 1 0.4362311 0.6186745
```

There are several ways to interpret this interval.

1. The process used to calculate this interval (take a random sample, calculate a statistic, repeatedly re-sample, and take the middle 95%) is a process that results in an interval that contains the parameter of interest on 95% of the samples we could have collected, however we don't know if the particular sample we collected and its resulting interval of (0.44, 0.62) is one of the intervals containing  $\mu$ .
2. We are 95% confident that  $\mu$  is in the interval (0.44, 0.62). This is delightfully vague and should be interpreted as a shorter version of the previous interpretation.
3. The interval (0.44, 0.62) is the set of values of  $\mu$  that are consistent with the observed data at the 0.05 threshold of statistical significance for a two-sided hypothesis test.

## 3.4 Additional Examples

### Example: Fuel Economy

Suppose we have data regarding fuel economy of 5 new vehicles of the same make and model and we wish to test if the observed fuel economy is consistent with the advertised 31 mpg at highway speeds. Here are the data:

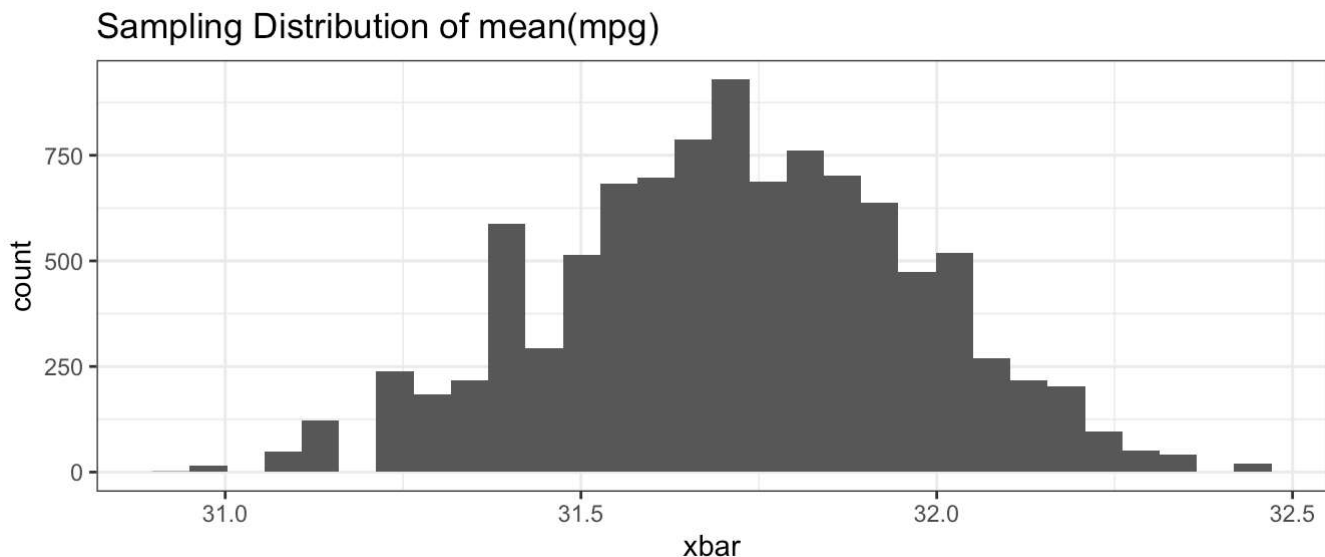
```
CarMPG <- data.frame( ID=1:5, mpg = c(31.8, 32.1, 32.5, 30.9, 31.3) )
CarMPG %>% summarise( xbar=mean(mpg) )

##      xbar
## 1 31.72
```

We will use the sample mean to assess if the sample fuel efficiency is consistent with the advertised number. Because these cars could be considered a random sample of all new cars of this make, we will create the estimated sampling distribution using the bootstrap re-sampling of the data.

```
# Run the bootstrap now with CarMPG$mpg as our data
BootDist <- boot(data = CarMPG$mpg, statistic = mean.function, R=10000)

# show a histogram of the sampling distribution of xbar
BootDist.graph <- data.frame(xbar=BootDist$t)
ggplot(BootDist.graph, aes(x=xbar)) +
  geom_histogram() +
  ggtitle('Sampling Distribution of mean(mpg)')
```



```
# calculate the 95% confidence interval using middle 95% of xbars
# confint( BootDist, level=.95, type='perc')
quantile( BootDist$t, probs=c(.025, .975) )

## 2.5% 97.5%
## 31.22 32.20
```

We see that the 95% confidence interval is (31.2, 32.2) and does not actually contain the advertised 31 mpg. However, I don't think we would object to a car manufacturer selling us a car that is better than advertised.

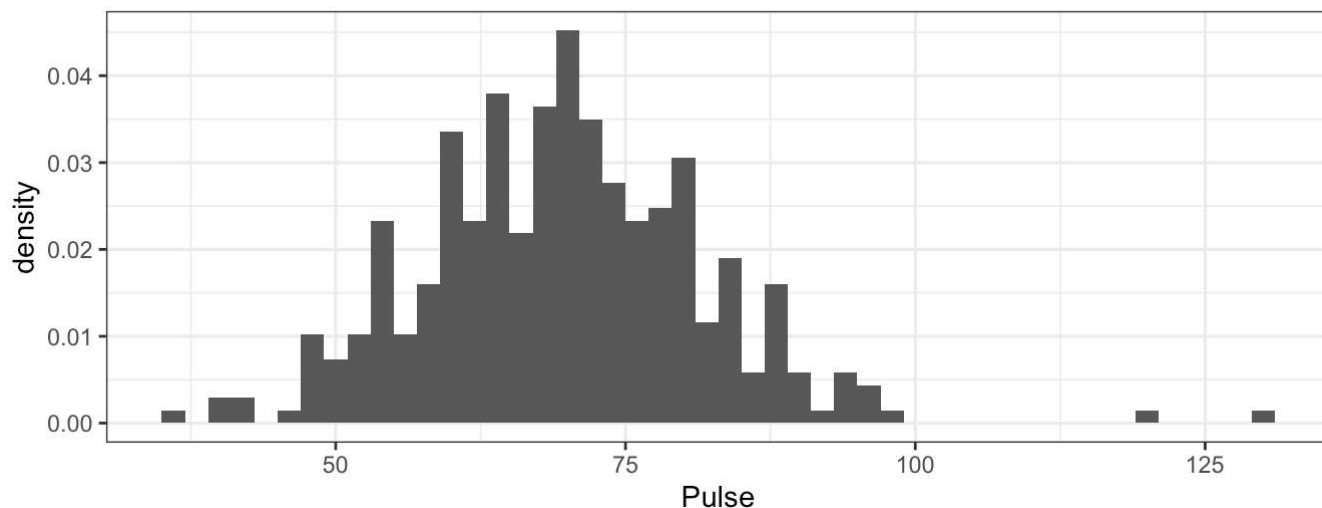
### Example: Pulse Rate of College Students

In the package `Lock5Data`, the dataset `GPAGender` contains information taken from undergraduate students in an Introductory Statistics course. This is a convenience sample, but could be considered representative of students at that university. One of the covariates measured was the students pulse rate and we will use this to create a confidence interval for average pulse of students at that university.

First we'll look at the raw data.

```
data(GPAGender, package='Lock5Data') # Load the dataset
# Now a nice histogram
ggplot(GPAGender, aes(x=Pulse, y=..density..)) +
  geom_histogram(binwidth=2) +
  ggtitle('Sample Data')
```

## Sample Data



It is worth noting this was supposed to be measuring resting heart rates, but there are two students had extremely high pulse rates and six with extremely low rates. The two high values are approximately what you'd expect from someone currently engaged in moderate exercise and the low values are levels we'd expect from highly trained endurance athletes.

```
# Summary Statistics
```

```
GPAGender %>% summarise(xbar = mean(Pulse),
                          StdDev = sd(Pulse))
```

```
##      xbar  StdDev
```

```
## 1 69.90379 12.08569
```

So the sample mean is  $\bar{x} = 69.9$  but how much should we expect our sample mean to vary from sample to sample when our sample size is  $n = 343$  people? We'll estimate the sampling distribution of  $\bar{X}$  using the bootstrap.

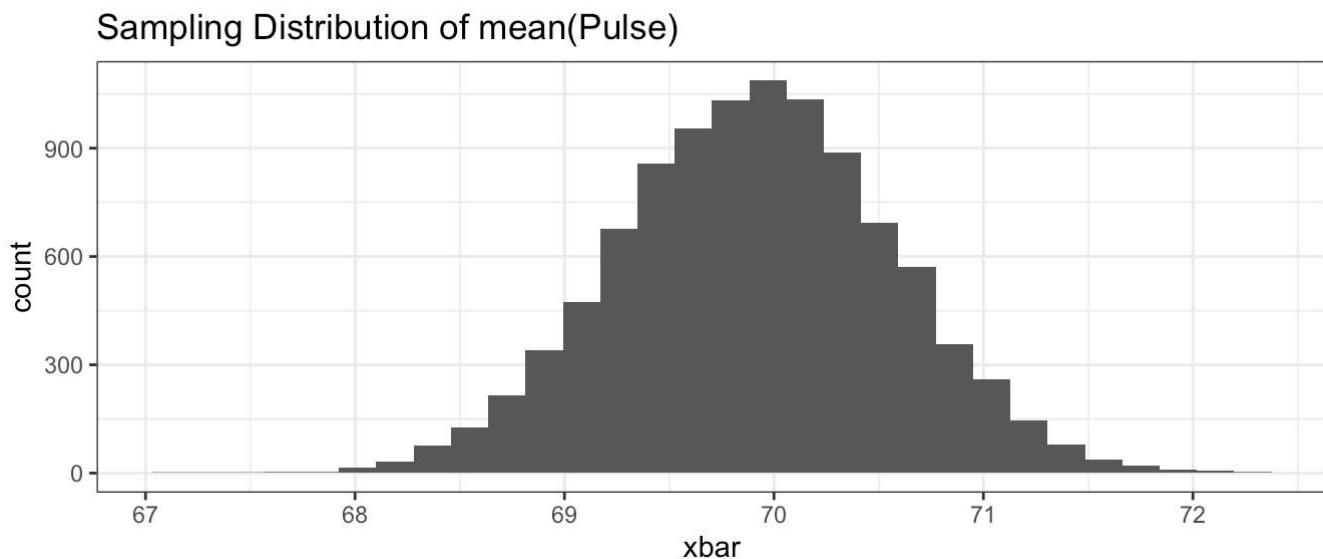
```
# Create the bootstrap replicates

BootDist <- boot(data = GPAGender$Pulse, statistic = mean.function, R=10000)

# show a histogram of the sampling distribution of xbar

BootDist.graph <- data.frame(xbar=BootDist$t)

ggplot(BootDist.graph, aes(x=xbar)) +
  geom_histogram() +
  ggtitle('Sampling Distribution of mean(Pulse)')
```



```
# confint( BootDist, Level=.95, type='perc')

quantile( BootDist$t, probs=c(.025, .975) )

##      2.5%      97.5%
## 68.63265 71.18375
```

Based on the quantile approach, the 95% bootstrap confidence for the mean pulse rate of undergraduates in the introductory statistics course is 68.7 to 71.2 beats per minutes.

## 3.5 Exercises

For several of these exercises, we will use data sets from the R package `Lock5Data`, which greatly contributed to the pedagogical approach of these notes. Install the package from CRAN using the RStudio point-and-click interface `Tools -> Install Packages ...`.

1. Load the dataset `BodyTemp50` from the `Lock5Data` package. This is a dataset of 50 healthy adults. One of the columns of this dataset is the `Pulse` of the 50 data points, which is the number of heartbeats per minute.
  - a. Create a histogram of the observed pulse values. Comment on the graph and aspects of the graph that might be of scientific interest. Below will help you load the data, and we want to use the `Pulse` variable.

```
data( BodyTemp50, package='Lock5Data' )
#?BodyTemp50
```

- b. Calculate the sample mean  $\bar{x}$  and sample standard deviation  $s$  of the pulses.
- c. Create a dataset of 10000 bootstrap replicates of  $\bar{x}^*$ .
- d. Create a histogram of the bootstrap replicates. Calculate the mean and standard deviation of this distribution. Notice that the standard deviation of the distribution is often called the Standard Error of  $\bar{x}$  and we'll denote it as  $\hat{\sigma}_{\bar{x}}$ .
- e. Using the bootstrap replicates, create a 95% confidence interval for  $\mu$ , the average adult heart rate.
- f. Calculate the interval

$$(\bar{x} - 2 \cdot \hat{\sigma}_{\bar{x}}, \bar{x} + 2 \cdot \hat{\sigma}_{\bar{x}})$$

and comment on its similarity to the interval you calculated in part (e).

2. Load the dataset `EmployedACS` from the `Lock5Data` package. This is a dataset drawn from American Community Survey results which is conducted monthly by the US Census Bureau and should be representative of US workers. The column `HoursWk` represents the number of hours worked per week.



- a. Create a histogram of the observed hours worked. Comment on the graph and aspects of the graph that might be of scientific interest.
- b. Calculate the sample mean  $\bar{x}$  and sample standard deviation  $s$  of the worked hours per week.
- c. Create a dataset of 10000 bootstrap replicates of  $\bar{x}^*$ .
- d. Create a histogram of the bootstrap replicates. Calculate the mean and standard deviation of this distribution. Notice that the standard deviation of the distribution is often called the Standard Error of  $\bar{x}$  and we'll denote it as  $\sigma_{\bar{x}}$ .
- e. Using the bootstrap replicates, create a 95% confidence interval for  $\mu$ , the average worked hours per week.
- f. Calculate the interval

$$(\bar{x} - 2 \cdot \hat{\sigma}_{\bar{x}}, \bar{x} + 2 \cdot \hat{\sigma}_{\bar{x}})$$

and comment on its similarity to the interval you calculated in part (e).

3. Return to the `BodyTemp50` data within the `Lock5Data` package, as in Exercise 1.

```
new.function <- function(x, index) {
  d <- x[index]
  return(mean(d)) } ## Hint: sd() calculates the standard deviation
```

- a. The code above was given in the chapter for calculate the mean of a vector. Modify the code below such that we can prepare a bootstrap confidence interval of the **standard deviation**,  $\sigma$ .
- b. Create a dataset of 10000 bootstrap replicates of  $\sigma^*$ .
- c. Using the bootstrap replicates, create a 95% confidence interval for  $\sigma$ , the standard deviation of adult heart rate.