

# Bootstrap Methods

Typically we use (asymptotic) theory to derive the sampling distribution of a statistic. From the sampling distribution, we can obtain the variance, construct confidence intervals, perform hypothesis tests, and more.

**Challenge:**

**Basic idea of bootstrapping:**

**“Bootstrap World”**

# 1 Nonparametric Bootstrap

Let  $Y_1, \dots, Y_n \sim F$  with pdf  $f(y)$ . Recall, the empirical cdf is defined as

Theoretical:

Bootstrap:

The idea behind the nonparametric bootstrap is to sample many data sets from  $F_n(y)$ , which can be achieved by resampling from the data **with replacement**.

```

# observed data
x <- c(2, 2, 1, 1, 5, 4, 4, 3, 1, 2)

# create 10 bootstrap samples
x_star <- matrix(NA, nrow = length(x), ncol = 10)
for(i in 1:10) {
  x_star[, i] <- sample(x, length(x), replace = TRUE)
}
x_star

```

```

##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]         1    2    4    1    2    1    2    3    3    4
## [2,]         4    4    1    1    1    2    2    1    2    1
## [3,]         2    2    2    4    5    4    4    5    1    4
## [4,]         4    4    2    5    2    4    5    5    1    3
## [5,]         2    1    5    1    3    2    4    2    4    4
## [6,]         4    4    2    1    4    4    4    3    1    2
## [7,]         1    1    2    1    2    1    2    2    3    1
## [8,]         4    4    1    3    3    3    5    1    2    4
## [9,]         4    1    2    3    2    1    2    1    4    2
## [10,]        3    4    5    1    5    4    5    2    4    1

```

```

# compare mean of the same to the means of the bootstrap samples
mean(x)

```

```
## [1] 2.5
```

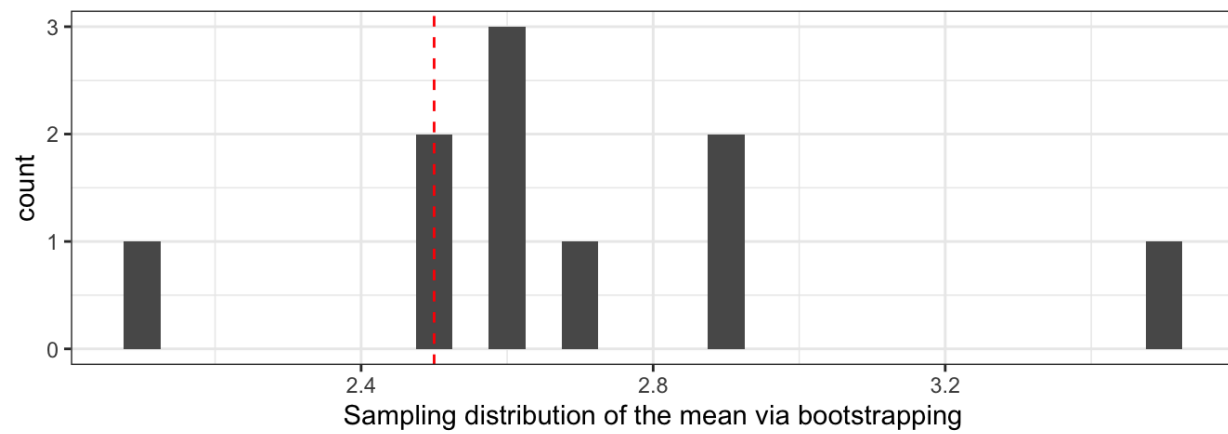
```
colMeans(x_star)
```

```
## [1] 2.9 2.7 2.6 2.1 2.9 2.6 3.5 2.5 2.5 2.6
```

```

ggplot() +
  geom_histogram(aes(colMeans(x_star)), binwidth = .05) +
  geom_vline(aes(xintercept = mean(x)), lty = 2, colour = "red") +
  xlab("Sampling distribution of the mean via bootstrapping")

```



## 1.1 Algorithm

**Goal:** estimate the sampling distribution of a statistic based on observed data  $x_1, \dots, x_n$ .

Let  $\theta$  be the parameter of interest and  $\hat{\theta}$  be an estimator of  $\theta$ . Then,

## 1.2 Justification for iid data

Suppose  $Y_1, \dots, Y_n$  are iid with  $\mathbf{E}Y_1 = \mu \in \mathbb{R}$ ,  $\mathbf{Var}(Y_1) = \sigma^2 \in (0, \infty)$ . Let's approximate the distribution of  $T_n = \sqrt{n}(\bar{Y}_n - \mu)$  via the bootstrap.

**Theorem:** If  $Y_1, Y_2, \dots$  are iid with  $\mathbf{Var}(Y_1) = \sigma^2 \in (0, \infty)$ , then  $\sup_{y \in \mathbb{R}} |P(T_n \leq y) - P_*(T_n^* \leq y)| \equiv \Delta_n \rightarrow 0$  as  $n \rightarrow \infty$  almost surely (a.s.).

The proof of this theorem requires two facts:

- i. (Berry-Esseen Lemma) Let  $Y_1, \dots, Y_n$  be independent with  $\mathbf{E}Y_i = 0$  and  $\mathbf{E}|Y_i|^3 < \infty$  for  $i = 1, \dots, n$ . Let  $\sigma_n^2 = n\mathbf{Var}(\bar{Y}_n) = n^{-1} \sum_{i=1}^n \mathbf{E}Y_i^2 > 0$ . Then,

$$\sup_{y \in \mathbb{R}} \left| P\left(\frac{\sqrt{n}\bar{Y}_n}{\sigma_n} \leq y\right) - \Phi(y) \right| = \sup_{x \in \mathbb{R}} \left| P(\sqrt{n}\bar{Y}_n \leq x) - \Phi\left(\frac{x}{\sigma_n}\right) \right| \leq \frac{2.75}{n^{3/2}\sigma_n^3} \sum_{i=1}^n \mathbf{E}|Y_i|^3.$$

- ii. (Marcinkiewicz-Zygmund SLLN) Let  $X_i$  be a sequence of iid random variables with  $\mathbf{E}|X_i|^p < \infty$  for  $p \in (0, 2)$ . Then, for  $S_n = \sum_{i=1}^n X_i$ ,

$$\frac{1}{n^{1/p}}(S_n - nc) \rightarrow 0 \text{ as } n \rightarrow \infty \text{ almost surely } (*)$$

for any  $c \in \mathbb{R}$  if  $p \in (0, 1)$  and for  $c = \mathbf{E}X_1$  if  $p \in [1, 2)$ . If  $(*)$  holds for some  $c \in \mathbb{R}$ , then  $\mathbf{E}|X_1|^p < \infty$ .



## 1.3 Properties of Estimators

We can use the bootstrap to estimate different properties of estimators.

### 1.3.1 Standard Error

Recall  $se(\hat{\theta}) = \sqrt{Var(\hat{\theta})}$ . We can get a **bootstrap** estimate of the standard error:

### 1.3.2 Bias

Recall  $bias(\hat{\theta}) = E[\hat{\theta} - \theta] = E[\hat{\theta}] - \theta$ . We can get a **bootstrap** estimate of the bias:

Overall, we seek statistics with small se and small bias.



## 1.4 Sample Size and # Bootstrap Samples

$n$  = sample size   &    $B$  = # bootstap samples

If  $n$  is too small, or sample isn't representative of the population,

Guidelines for  $B$  –

Best approach –

## Your Turn

In this example, we explore bootstrapping in the rare case where we know the values for the entire population. If you have all the data from the population, you don't need to bootstrap (or really, inference). It is useful to learn about bootstrapping by comparing to the truth in this example.

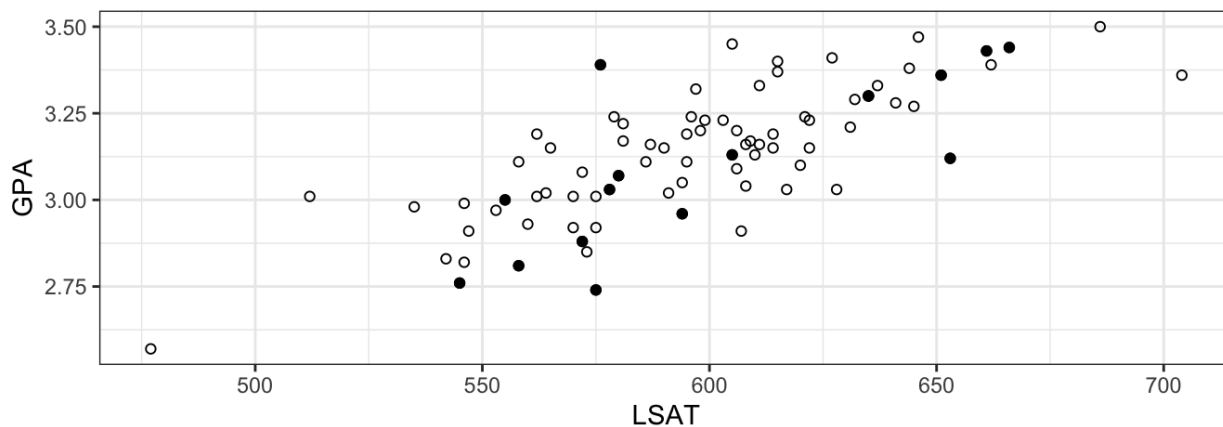
In the package `bootstrap` is contained the average LSAT and GPA for admission to the population of 82 USA Law schools (an old data set – there are now over 200 law schools). This package also contains a random sample of size  $n = 15$  from this dataset.

```
library(bootstrap)
```

```
head(law)
```

```
##   LSAT  GPA
## 1  576 3.39
## 2  635 3.30
## 3  558 2.81
## 4  578 3.03
## 5  666 3.44
## 6  580 3.07
```

```
ggplot() +
  geom_point(aes(LSAT, GPA), data = law) +
  geom_point(aes(LSAT, GPA), data = law82, pch = 1)
```



We will estimate the correlation  $\theta = \rho(\text{LSAT}, \text{GPA})$  between these two variables and use a bootstrap to estimate the sample distribution of  $\hat{\theta}$ .

```
# sample correlation  
cor(law$LSAT, law$GPA)
```

```
## [1] 0.7763745
```

```
# population correlation  
cor(law82$LSAT, law82$GPA)
```

```
## [1] 0.7599979
```

```
# set up the bootstrap  
B <- 200  
n <- nrow(law)  
r <- numeric(B) # storage  
  
for(b in B) {  
  ## Your Turn: Do the bootstrap!  
}
```

1. Plot the sample distribution of  $\hat{\theta}$ . Add vertical lines for the true value  $\theta$  and the sample estimate  $\hat{\theta}$ .
2. Estimate  $sd(\hat{\theta})$ .
3. Estimate the bias of  $\hat{\theta}$

## 1.5 Bootstrap CIs

We will look at five different ways to create confidence intervals using the bootstrap and discuss which to use when.

1. Percentile Bootstrap CI
2. Basic Bootstrap CI
3. Standard Normal Bootstrap CI
4. Bootstrap  $t$
5. Accelerated Bias-Corrected (BCa)

**Key ideas:**

### 1.5.1 Percentile Bootstrap CI

Let  $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$  be bootstrap replicates and let  $\hat{\theta}_{\alpha/2}$  be the  $\alpha/2$  quantile of  $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$ .

Then, the  $100(1 - \alpha)\%$  Percentile Bootstrap CI for  $\theta$  is

In R, if `bootstrap.reps = c( $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$ )`, the percentile CI is

```
quantile(bootstrap.reps, c(alpha/2, 1 - alpha/2))
```

**Assumptions/usage**

### 1.5.2 Basic Bootstrap CI

The  $100(1 - \alpha)\%$  Basic Bootstrap CI for  $\theta$  is

**Assumptions/usage**

### 1.5.3 Bootstrap $t$ CI (Studentized Bootstrap)

Even if the distribution of  $\hat{\theta}$  is Normal and  $\hat{\theta}$  is unbiased for  $\theta$ , the Normal distribution is not exactly correct for  $z$ .

Additionally, the distribution of  $\hat{se}(\hat{\theta})$  is unknown.

⇒ The bootstrap  $t$  interval does not use a Student  $t$  distribution as the reference distribution, instead we estimate the distribution of a “ $t$  type” statistic by resampling.

The  $100(1 - \alpha)\%$  Bootstrap  $t$  CI is

#### Overview

To estimate the “ $t$  style distribution” for  $\hat{\theta}$ ,

**Assumptions/usage**



### 1.5.4 BCa CIs

Modified version of percentile intervals that adjusts for bias of estimator and skewness of the sampling distribution.

This method automatically selects a transformation so that the normality assumption holds.

**Idea:**

The BCa method uses bootstrapping to estimate the bias and skewness then modifies which percentiles are chosen to get the appropriate confidence limits for a given data set.

**In summary,**

## Your Turn

We will consider a telephone repair example from Hesterberg (2014). `verizon` has repair times, with two groups, CLEC and ILEC, customers of the “Competitive” and “Incumbent” local exchange carrier.

```
library(resample) # package containing the data
```

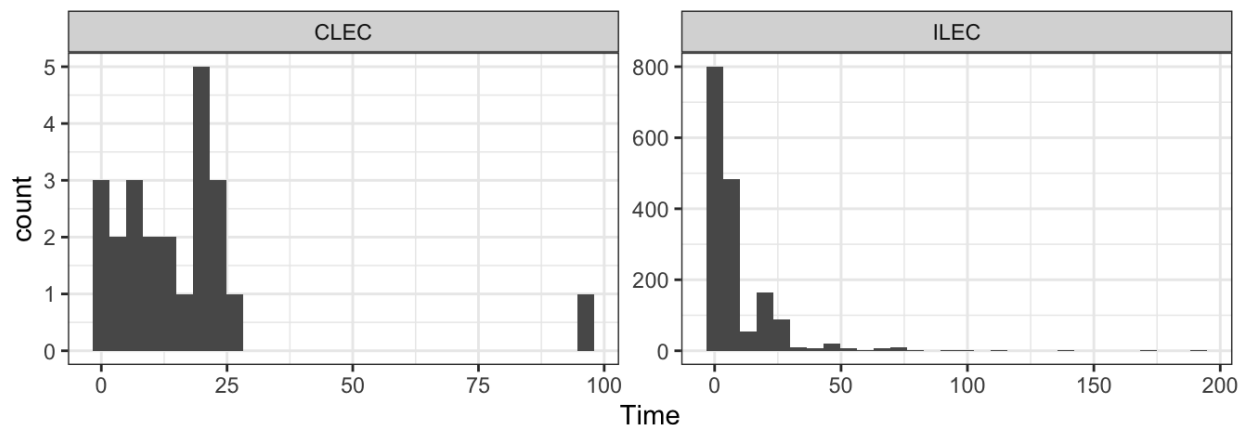
```
data(Verizon)
head(Verizon)
```

```
##      Time Group
## 1 17.50  ILEC
## 2  2.40  ILEC
## 3  0.00  ILEC
## 4  0.65  ILEC
## 5 22.23  ILEC
## 6  1.20  ILEC
```

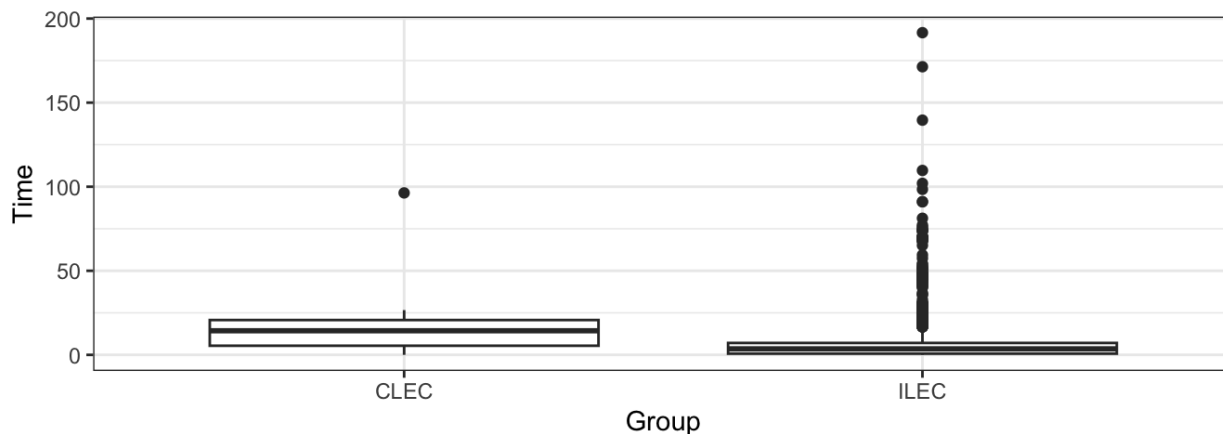
```
Verizon |>
  group_by(Group) |>
  summarize(mean = mean(Time), sd = sd(Time), min = min(Time), max =
            max(Time)) |>
  kable()
```

Group	mean	sd	min	max
CLEC	16.509130	19.50358	0	96.32
ILEC	8.411611	14.69004	0	191.60

```
ggplot(Verizon) +
  geom_histogram(aes(Time)) +
  facet_wrap(~Group, scales = "free")
```



```
ggplot(Verizon) +
  geom_boxplot(aes(Group, Time))
```



## 1.6 Bootstrapping CIs

There are many bootstrapping packages in **R**, we will use the `boot` package. The function `boot` generates  $R$  resamples of the data and computes the desired statistic(s) for each sample. This function requires 3 arguments:

1. `data` = the data from the original sample (data.frame or matrix).
2. `statistic` = a function to compute the statistic from the data where the first argument is the data and the second argument is the indices of the observations in the bootstrap sample.
3.  $R$  = the number of bootstrap replicates.

```

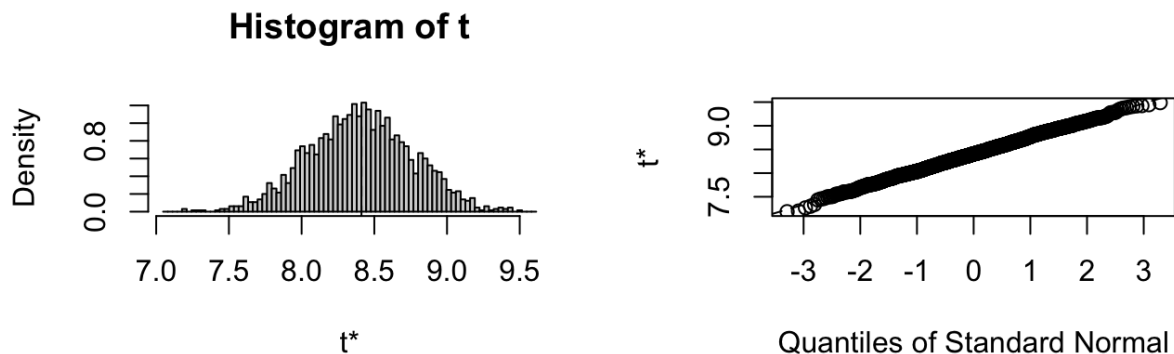
library(boot) # package containing the bootstrap function

mean_func <- function(x, idx) {
  mean(x[idx])
}

ilec_times <- Verizon[Verizon$Group == "ILEC",]$Time
boot.ilec <- boot(ilec_times, mean_func, 2000)

plot(boot.ilec)

```



If we want to get Bootstrap CIs, we can use the `boot.ci` function to generate the different nonparametric bootstrap confidence intervals.

```

boot.ci(boot.ilec, conf = .95, type = c("perc", "basic", "bca"))

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.ilec, conf = 0.95, type = c("perc",
## "basic",
## "bca"))
##
## Intervals :
## Level      Basic          Percentile          BCa
## 95%    ( 7.733,  9.110 )  ( 7.714,  9.091 )  ( 7.755,  9.125 )
## Calculations and Intervals on Original Scale

```

```
## we can do some of these on our own
## percentile
quantile(boot.ilec$t, c(.025, .975))
```

```
##      2.5%      97.5%
## 7.714075 9.084725
```

```
## basic
2*mean(ilec_times) - quantile(boot.ilec$t, c(.975, .025))
```

```
##      97.5%      2.5%
## 7.738496 9.109147
```

To get the studentized bootstrap CI, we need our statistic function to also return the variance of  $\hat{\theta}$ .

```
mean_var_func <- function(x, idx) {
  c(mean(x[idx]), var(x[idx])/length(idx))
}

boot.ilec_2 <- boot(ilec_times, mean_var_func, 2000)
boot.ci(boot.ilec_2, conf = .95, type = "stud")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.ilec_2, conf = 0.95, type = "stud")
##
## Intervals :
## Level      Studentized
## 95%      ( 7.728,  9.183 )
## Calculations and Intervals on Original Scale
```

Which CI should we use?

## 1.7 Bootstrapping for the difference of two means

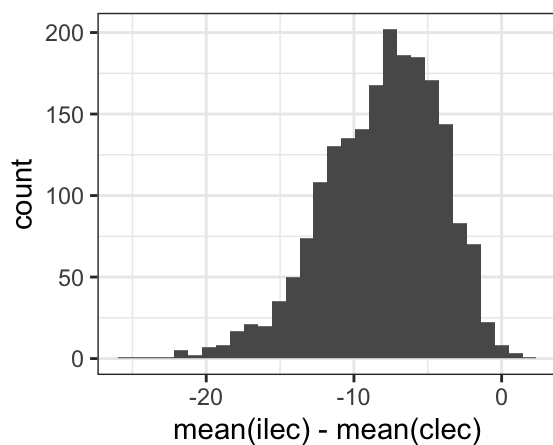
Given iid draws of size  $n$  and  $m$  from two populations, to compare the means of the two groups using the bootstrap,

The function `two.boot` in the `simpleboot` package is used to bootstrap the difference between univariate statistics. Use the bootstrap to compute the shape, bias, and bootstrap sample error for the samples from the Verizon data set of CLEC and ILEC customers.

```
library(simpleboot)

clec_times <- Verizon[Verizon$Group == "CLEC",]$Time
diff_means.boot <- two.boot(ilec_times, clec_times, "mean", R = 2000)

ggplot() +
  geom_histogram(aes(diff_means.boot$t)) +
  xlab("mean(ilec) - mean(clec)")
```



```
# Your turn: estimate the bias and se of the sampling distribution
```

Which confidence intervals should we use?

```
# Your turn: get the chosen CI using boot.ci
```

Is there evidence that

$$H_0 : \mu_1 - \mu_2 = 0$$

$$H_a : \mu_1 - \mu_2 < 0$$

is rejected?