Bootstrap Confidence Interval

Neba Nfonsang
University of Denver

In [1]:

```
require(tidyverse)
```

```
Loading required package: tidyverse
Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr
Conflicts with tidy packages ---------------------------------------
-------------
filter(): dplyr, stats
lag():    dplyr, stats
```

What is Bootstrapping?

- Bootstrapping is a resampling technique used to estimate an unknown population parameter by repeatedly sampling from the entire data. It is a more restricted form of resampling because the samples are drawn from the data and not from the population. We do not have any information about the population in this situation. So, the assumption is that, the data we are sampling from looks like the population or is representative of the population.

Estimation Problem

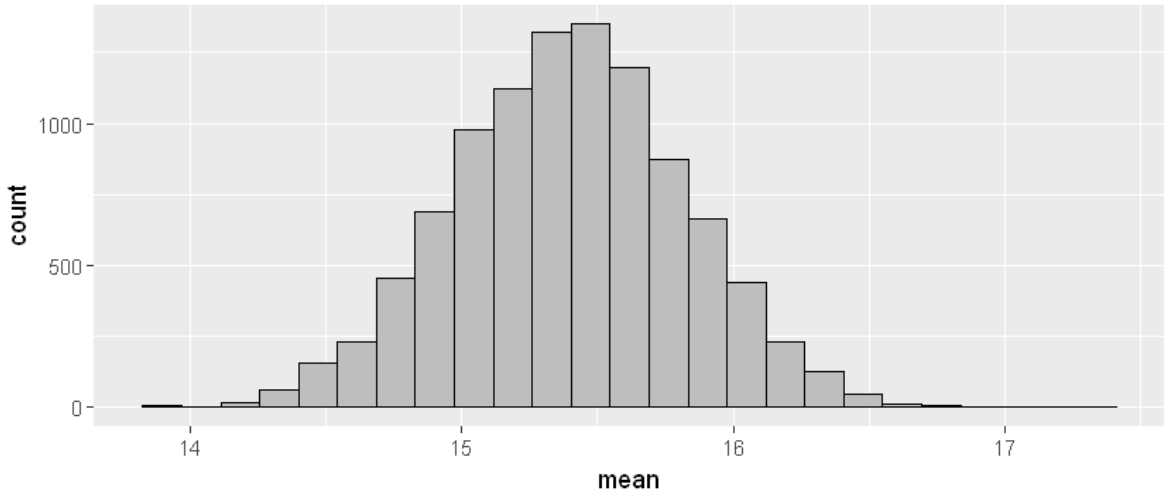
- The data points are percentages of aluminum content in a sample of 18 ancient Theban jars (Desphpande et. al., 1996, p. 31)
- Estimate the mean of the population from which the data came

In [2]:

```
data = c( 11.4, 13.4, 13.5, 13.8, 13.9,14.4, 14.5, 15.0,
          15.1, 15.8, 16.0, 16.3, 16.5, 16.9, 17.0, 17.2,
          17.5, 19.0)
```

In [3]:

```
# bootstrapping

set.seed(1)

iterations = 10000
sim.values <- rep(NA, iterations)

for (i in 1:iterations){
    # sample from the data with replacement
    samp = sample(data, size=length(data), replace=T)

    # record the mean of each sample
    sim.values[i] = mean(samp)
}

dat = data.frame(mean = sim.values)

options(repr.plot.height=3)
ggplot(data=dat, aes(x=mean)) +
geom_histogram(bins=25, fill="grey", color="black")
```



Compute the 95% Confidence Interval

In [4]:

```
# 95% confidence interval
quantile(sim.values, c(0.025, 0.975))
```

2.5%
14.5444444444444
97.5%
16.2222222222222

In [5]:

```
# calculate the 95% confidence interval using percentile

## formula: index.position = q(n + 1)
## pass the index.position into the sorted data

ind = c(0.025, 0.975)*(length(sim.values) + 1)
ind = round(ind)

sort(sim.values)[ind]
```

14.5444444444444 16.2222222222222

- A conventional approach uses the t-test to get the confidence interval

In [6]:

```
# extract confidence interval from t-test results
t.test(data)$conf
```

14.4779830924965 16.3220169075035

Hypothesis Testing Problem

- Bootstrapping can also be applied to hypothesis testing where the confidence interval approach is used.

- Is there a statistical significant difference between the following samples?
- That is, do the two samples x1 and x2 come from different populations?
- The null hypothesis is that, the means of the populations from which the samples were drawn are equal (mu1 = mu2)
- In other words, the difference in the means of the populations from which the samples were collected is zero (mu1 - mu2 = 0).

This question can be answered through resampling using the following steps:

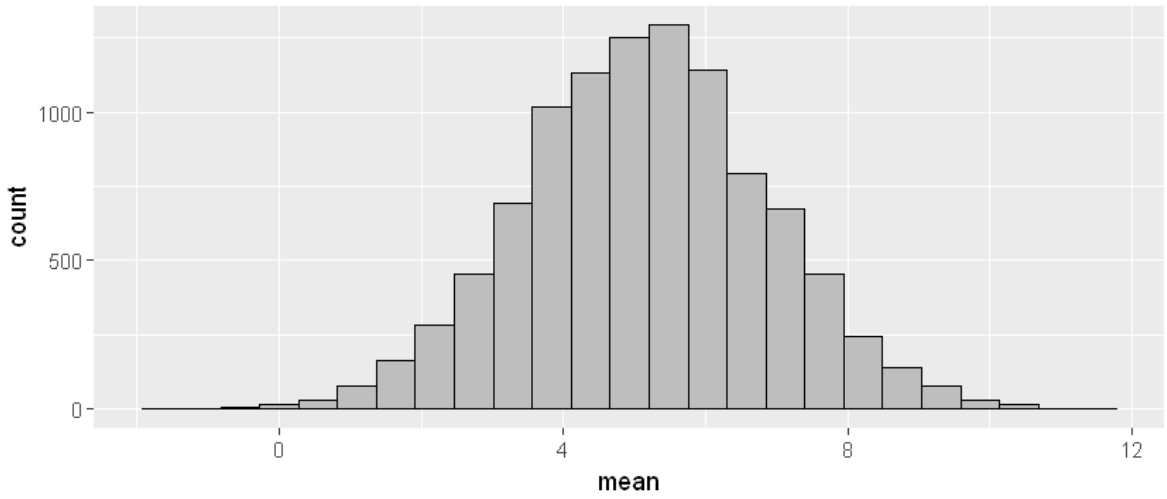
- a sample is drawn from x1 (with replacement) and its mean is computed
- a sample is drawn from x2 (with replacement) and its mean is computed
- the difference between the two means are computed and recorded (tracked)
- a 95% confidence interval is then calculated for the simulated mean differences
- The results are interpreted: if the confidence interval includes the hypothesized mean difference of zero, then we fail to reject the null hypothesis and conclude that the samples were drawn from the same population. Otherwise, we conclude that the samples were collected from different populations.

In [7]:

```
x1 = c(52, 48, 49, 58, 52, 55, 45, 51, 52,57,
       43, 44, 42, 41, 58, 44, 57, 45, 44, 49)

x2 = c(50, 55, 61, 53, 54, 52, 63, 63, 50, 59,
       56, 63, 56, 44, 55, 57, 46, 50, 54, 48)
```

In [8]:

```
set.seed(1)
iterations = 10000
sim.mean.diff <- rep(NA, iterations)

for (i in 1:iterations){
    samp1 = sample(x1, size=length(x1), replace=T)
    samp2 = sample(x2, size=length(x2), replace=T)

    sim.mean.diff[i] = mean(samp2) - mean (samp1)
}

dat = data.frame(mean = sim.mean.diff)

options(repr.plot.height=3)
ggplot(data=dat, aes(x=mean)) +
geom_histogram(bins=25, fill="grey", color="black")
```



In [9]:

```
# 95% confidence interval
quantile(sim.mean.diff, c(0.025, 0.975))
```

2.5%
1.79875
97.5%
8.59999999999999

- It can be concluded that the two samples came from different populations since the hypothesized mean difference of 0 is not included in the 95% percent confidence interval.

Verify the Bootstrap Hypothesis Test Results Using a T-Test

In [10]:

```
t.test(x2, x1)
```

```
        Welch Two Sample t-test

data:  x2 and x1
t = 2.9015, df = 37.98, p-value = 0.006148
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.556771 8.743229
sample estimates:
mean of x mean of y
    54.45     49.30
```

- The p-value is less than 0.05 indicating that the samples are more likely to come from different populations.

- You would also notice that the confidence interval could be gotten from this t-test as well:

In [11]:

```
# get confidence interval from t-test
t.test(x2, x1)$conf
```

1.55677053043876 8.74322946956126

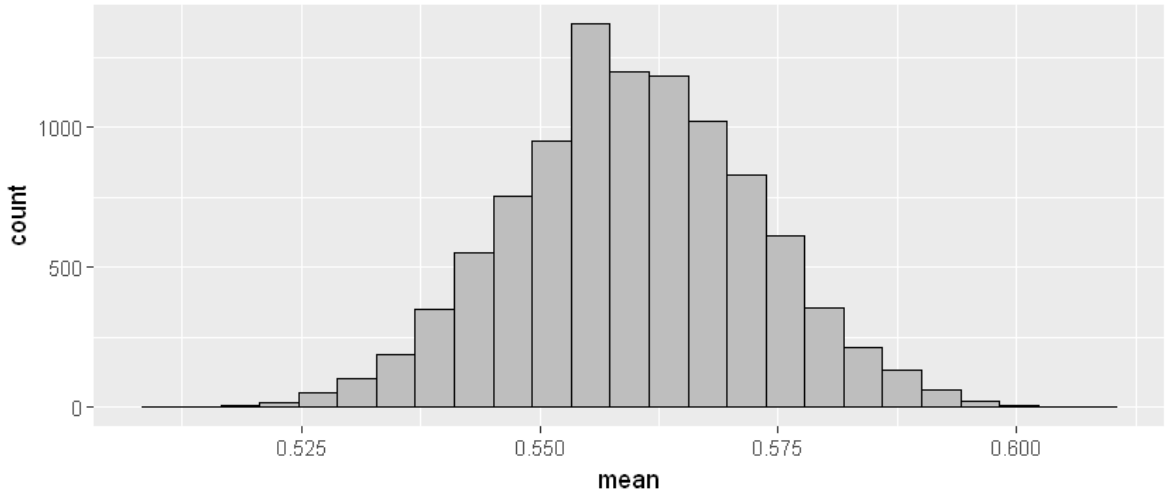
Confidence Interval Estimate for Proportion Data

Problem

- Consider the data of randomly selected 1988 presidential election poll, showing 840 intended votes for Bush and 660 intended votes for Dukakis out of 1500 (Wonnacott and Wonnacott, 1990, p. 5).
- What is the 95% confidence interval estimate for the proportion of Bush supporters in the population?

In [12]:

```
# bootstrapping

set.seed(1)


iterations = 10000
sim.prop <- rep(NA, iterations)

for (i in 1:iterations){
    # sample from the data with replacement
    samp = sample(c(0, 1), size=1500, replace=T, prob=c(660/1500, 840/1500))

    # record the mean of each sample
    sim.prop[i] = mean(samp)
}

dat = data.frame(mean = sim.prop)

options(repr.plot.height=3)
ggplot(data=dat, aes(x=mean)) +
geom_histogram(bins=25, fill="grey", color="black")
```



In [13]:

```
# confidence interval estimate for the proportion of Bush supporters
# in the population
quantile(sim.prop, c(0.025, 0.975))
```

2.5%
0.534666666666667
97.5%
0.585333333333333