STA258H5

Al Nosedal
and Alison Weir

Winter 2017

BOOTSTRAP CONFIDENCE INTERVALS

Distribution of Sample Mean

Let X_i for $i = 1, 2, ..., n$ be a sample of iid random variables with $E(X_i) = \mu$ and $Var(X_i) = \sigma^2$. The sample mean is defined by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

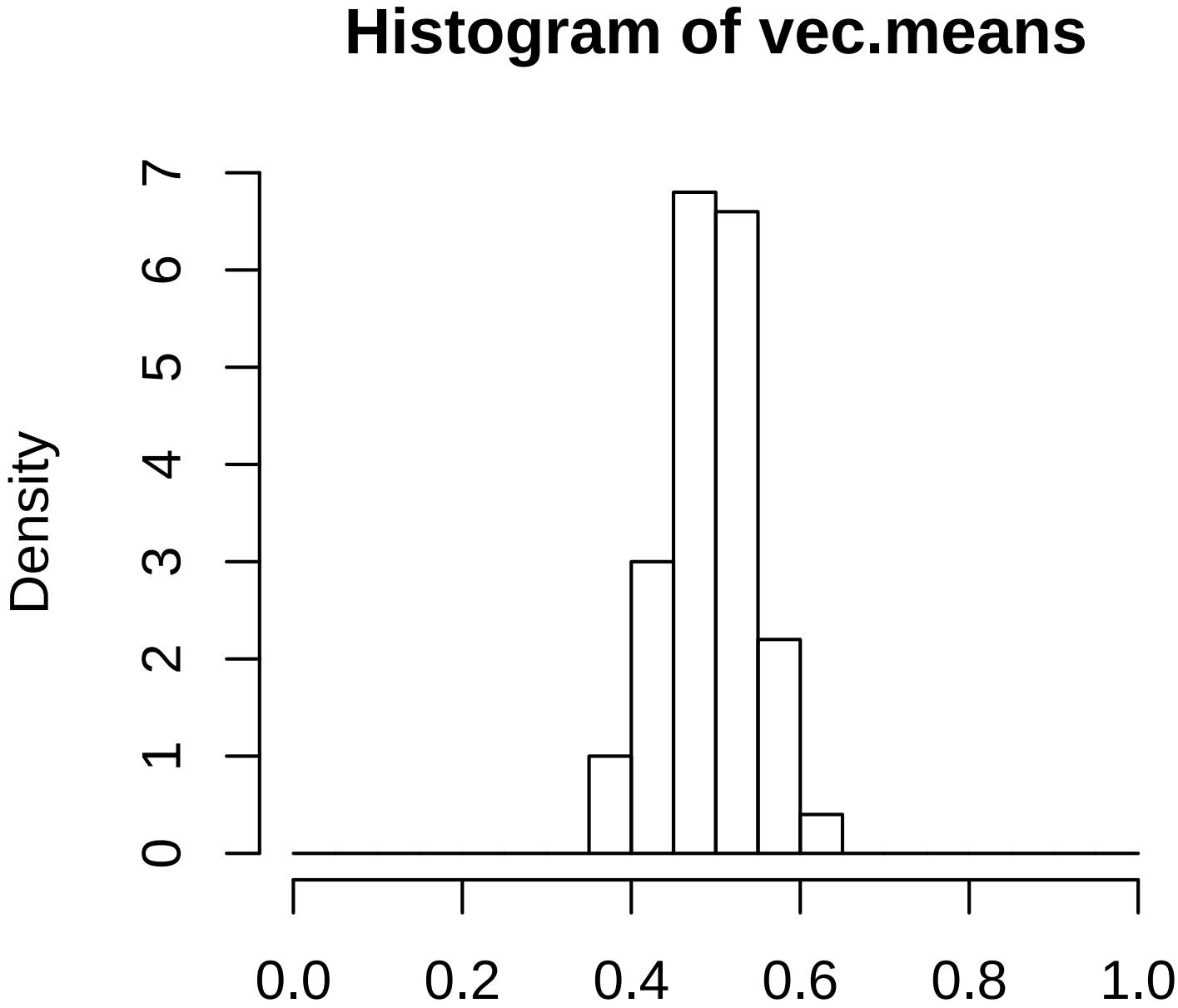
Using the properties of expected value, it is easy to show that $E(\bar{X}) = \mu$ and $Var(\bar{X}) = \frac{\sigma^2}{n}$, Right?

Example

Here we draw a list of 25 uniformly distributed random numbers, compute the mean, and repeat this 100 times. This will give us 100 different estimates of the mean of the underlying distribution.

R Code

```r
# vector of means = vec.means

vec.means<-matrix(0,nrow=100,ncol=1)

for (i in 1:100){

        vec.means[i,1]<-mean( runif(25)  )


}

bins<-seq(0,1,by=0.05)

hist(vec.means,prob=TRUE,breaks=bins)
```


Histogram of vec.means
Density
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0

Example (cont.)

Let us look at the distribution of these 100 calculated means; this histogram can be viewed as an estimate of the true sampling distribution.

Example (cont.)

From our table, we have that

$$E(\bar{X}) = 0.5$$

and

$$V(\bar{X}) = \frac{1/12}{25} \approx 0.003333$$

The Bootstrap (cont.)

Unfortunately, in most cases we do not know the underlying distribution from which the sample is drawn. At best we may suspect that the true distribution is in some family of distributions, but we generally do not know the parameters of the distribution.
So suppose that we have just one sample. Is there any way to use that one sample to compute an estimate of the sampling distribution of a statistic? This is where the bootstrap comes in.

The Bootstrap (cont.)

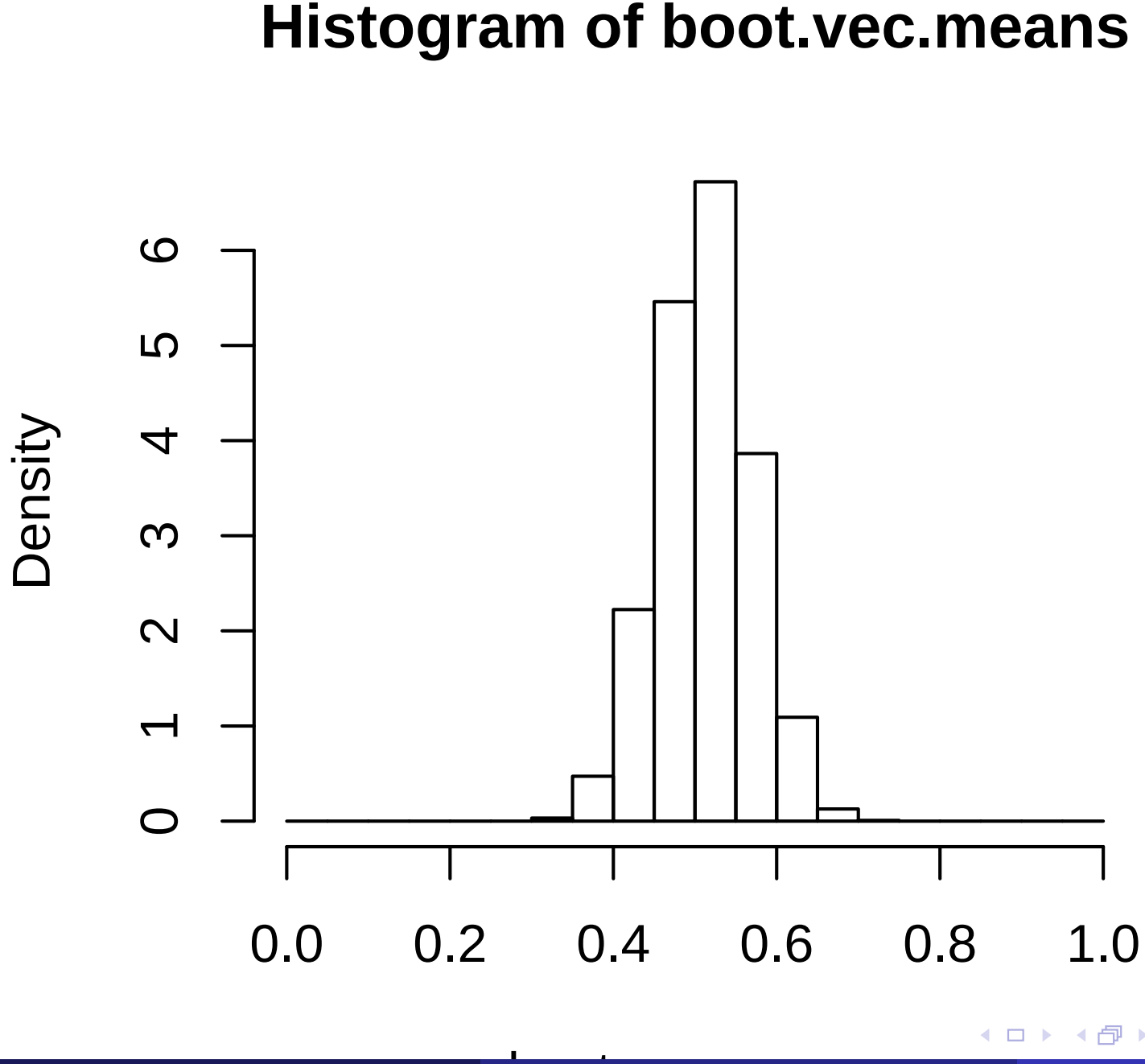
The idea is to repeatedly sample (with replacement) from the single sample you have, and use these "samples" to compute the sampling distribution of the statistic in which you are interested. If our original sample is reasonably representative of the population, then resampling from that sample should look pretty much like drawing a new sample.

Resampling

```
### original = original sample;
original.sample<-seq(1,6,by=1);

original.sample

## [1] 1 2 3 4 5 6

sample(original.sample,replace=TRUE);

## [1] 1 2 2 6 5 5
```

R Code (Bootstrap Sample)

```
### original = original sample;
original<-runif(25);

# vector of means = vec.means
boot.vec.means<-matrix(0,nrow=5000,ncol=1)
for (i in 1:5000){
boot.vec.means[i,1]<-mean( sample(original,replace=TRUE) )
                              }

bins<-seq(0,1,by=0.05)
hist(boot.vec.means,prob=TRUE,breaks=bins)
```


Histogram of boot.vec.means
Density
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0

Summary statistics (bootstrap distribution)

```
# estimate of population mean;
mean(boot.vec.means);

## [1] 0.5125034

# estimate of variance of x bar;
var(boot.vec.means);

##               [,1]
## [1,] 0.003183633
```

Note that the distributions are reasonably similar. This is the genius of the bootstrap: resampling from the single sample provides a reasonable way to estimate what would happen if we actually drew many separate samples.

The empirical bootstrap

Suppose we have n data points $x_1, x_2, ..., x_n$ drawn from a distribution F. An **empirical bootstrap sample** is a resample of the same size n: $x_1^*, x_2^*, ..., x_n^*$.
You should think of the latter as a sample of size n drawn from the empirical distribution F^*. For any statistic v computed from the original sample data, we can define a statistic v^* by the same formula but computed instead using the resampled data. With this notation we can state the bootstrap principle:

- $F^* \approx F$.
- The distribution of v^* approximates the distribution of v.

The empirical bootstrap (cont.)

It turns out that in practice the approximation of v by v^* may not be very good. However, the variation of v is usually well-approximated by the variation of v^*.

Toy example of an empirical bootstrap confidence interval

The sample data is $30, 37, 36, 43, 42, 43, 43, 46, 41, 42$.
Estimate the mean μ of the underlying distribution and give an 80% confidence interval.

Solution

We use $\bar{x} = 40.3$ to estimate the true mean μ of the underlying distribution. To make the confidence interval we need to know how much the distribution of $\bar{x}$ varies around μ. That is, we would like to know the distribution of

$$\delta = \bar{x} - \mu.$$

Solution

If we knew this distribution we could find $\delta_{0.1}$ and $\delta_{0.9}$ critical values of δ. Then we would have

$$P(\delta_{0.1} < \bar{x} - \mu < \delta_{0.9}) = 0.8$$

which is equivalent to

$$P(\bar{x} - \delta_{0.9} < \mu < \bar{x} - \delta_{0.1}) = 0.8$$

Solution

The bootstrap principle offers a practical approach to estimating the distribution of $\delta = \bar{x} - \mu$. It says that we can approximate it by the distribution of

$$\delta^* = \bar{x}^* - \bar{x}$$

where $\bar{x}^*$ is the mean of an empirical bootstrap sample.

R Code

```
set.seed(686);

# x = data;
x <- c(30,37,36,43,42,43,43,46,41,42);
n <- length(x);
xbar<-mean(x);
# nboot = bootstrap samples;
nboot <-20;
resamples <- sample(x,n*nboot, replace = TRUE);
boot_sample<-matrix(resamples, nrow=n, ncol=nboot);
```

R Code

```
# xbar.star = bootstrap sample means;
xbar.star <- colMeans(boot_sample);
delta.star<-xbar.star - xbar;
# order results;
new.delta.star<-sort(delta.star);

# Find quantiles
d9 <-delta.star[18];
d1<-delta.star[2];
CI<-xbar-c(d9,d1);
print(CI);

## [1] 39.1 41.5
```

Another example

```
#Step 0. Install R Package boot;

library(boot);

#Step 1. Read data;
glass<-read.csv(file="glass.csv", header=TRUE);
names(glass);

##  [1] "id"   "ri"   "na"   "mg"   "al"   "si"   "k"    "ca"
## [11] "type"
```

Another example

```
#Step 2. Create Bootstrap Sample;

y<-glass$ri;

mean.fn<-function(y,id){mean(y[id])}

boot.out<-boot(y,mean.fn,R=2000)

boot.out$t[1:5]

## [1] 1.518274 1.517992 1.518184 1.518347 1.517809
```

Another example (Basic)

```
#Step 3. Construct CI;

boot.ci(boot.out,type="basic");

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.out, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   ( 1.518,  1.519 )
## Calculations and Intervals on Original Scale
```

Another example (Percentile)

```
#Step 3. Construct CI;

boot.ci(boot.out,type="perc");

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.out, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   ( 1.518,  1.519 )
## Calculations and Intervals on Original Scale
```