Sampling Variability, Confidence Intervals and the Bootstrap

Data Science Team

Brief Recap: What have we learned so far?

- Often the data is a sample from a population and we want to use it to learn something about this bigger population. Sometimes the larger population is defined by a concrete number of subjects, sometimes an abstraction.
- When the sample size is limited, there is a considerable variability of the sample statistic in repeated sampling.
- A sample statistic is a reasonable estimate for the corresponding population statistic (as the sample size grows bigger, the sample statistic becomes very close to the population value).
- Today: We want to understand how “off” our estimate from the sample will be from the true population parameter.

Example: Stanford degrees conferred

- Stanford keeps track of all of the degrees it confers in each year. What proportion of the degrees awarded in academic year 2016-2017 were Bachelor's degrees?
- In this case (we know the entire population), we can just look at this table:

```
library(readr)
Stanford <- read_csv("data/stanford_degrees.csv")
names(Stanford) <- c("DegreeCategory","Number")
Stanford
```

```
# A tibble: 4 x 2
  DegreeCategory                            Number
  <chr>                                      <dbl>
1 Bachelor's degrees                          1669
2 Master's degrees                            2406
3 Doctoral degrees - research/scholarship      752
4 Doctoral degrees - professional practice     271
```

Terminology

Recall our terminology:

- **Population**: All degrees awarded at Stanford in 2016-2017.
- **Parameter**: Proportion of Bachelor Degrees.

In this case, the whole population is tabulated (previous slide), so we can easily compute the parameter: 0.327.

Typically, the population and the parameter are unknown. Then we estimate the parameter with a sample. Suppose we sample 50 degrees at random.

- **Statistic (Estimate)**: Proportion of Bachelor degrees in the sample.

The key question is: How much is this estimate off?

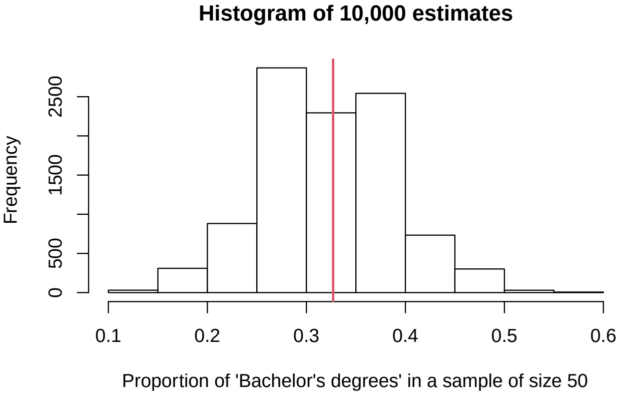
Bias and Chance Error

- Since the sample is drawn at random, the estimate will be different from the parameter. Drawing another sample will result in a different random (chance) error.
- There might also be some **bias** (*systematic* error), so Estimate = Parameter + Bias + Random Error.
- We will soon see that the chance error will get smaller as the sample size gets bigger. Moreover, we can compute how large the chance error will be.
- This is not the case for the bias: Increasing the sample size just repeats this systematic error on a larger scale.
- We'll focus on the case where we don't have bias.

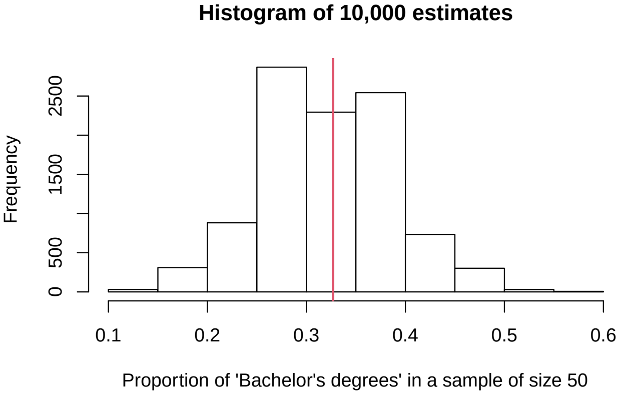
Estimating a parameter

The parameter is the proportion of Bachelor degrees in the whole population: 0.327 (red line). If we estimate this with a sample of size 50, we will be off somewhat.

- If we repeat the sampling 10,000 times, calculating the proportion of Bachelor Degrees in each sample of size 50, we get 10,000 estimates:



The sampling distribution



This is called the **sampling distribution of the statistic**. It is a theoretical construct, because it is the histogram of infinitely many estimates (although it can be approximated with 10,000 estimates as above.)

How much are we off?

The sampling distribution shows that there is quite some variability in the estimate: The standard deviation of the sampling distribution is:

```
SE <- sqrt(var(SamplePropB))
print(SE)
```

```
           [,1]
[1,] 0.06661393
```

This is called the **standard error (SE)** of our estimate. It tells us how far we can expect the estimate to be from the parameter.

The square root law

Suppose $\bar{X}$ is the average of n independent observations from a population with standard deviation σ. Then:

$$\mathrm{SE}(\bar{X}) = \frac{\sigma}{\sqrt{n}}$$

The importance of the square root law is twofold:

- It shows that the SE becomes smaller if we use a larger sample size n. We can use the formula to determine what sample size is required for a desired accuracy.
- The SE **does not depend on the size of the population**, only on the size of the sample.

Recap: The Sample Average

Setup:

- We have a population with mean μ and variance σ^2.
- Let $(X_1, \ldots, X_n)$ be a random sample of size n from that population.
- Let $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$ be the sample mean.

What we have seen is that if our estimate is an average, then:

- The **average value of $\bar{X}$ across all the possible samples** we might take is equal to μ.
- The standard error is $SE = \frac{\sigma}{\sqrt{n}}$ (or the **variance of $\bar{X}$ across all the possible samples** is equal to σ^2/n).

The Sample Average and the CLT

- Sums (or averages) of many independent random variables all with the same mean have approximately a Normal sampling distribution (this is known at the Central Limit Theorem (CLT)).

- For example, suppose $\bar{X}$ is the average of n independent observations from a population with mean μ and standard deviation σ.

- For a sample size n "big enough," the **histogram of $\bar{X}$ across all possible samples** has the shape of a Normal distribution

$$\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$$

Estimating the SE for an average

- Recall that σ in $\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$ is the standard deviation of the population!
- Now we have a problem: We don't know σ because we don't know all of the data in the population (which is why we took a sample in the first place)!.
- Plug-in method: Estimate σ by its sample version s (the standard deviation in the sample) .

Example

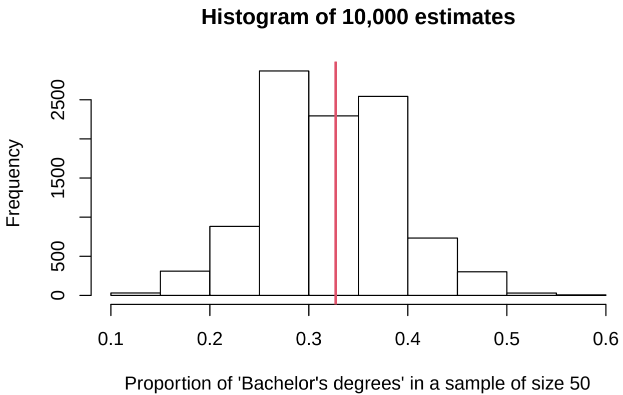
- Consider a poll on the approval rating of the US President. Suppose 60% of 140 million likely voters approve of the way the president is handling his job.We poll 1,000 of them. The resulting approval percentage in the sample will be off the population percentage of 60% due to random / chance error.
- Suppose that the percentage in our sample was 58%. We want to know how off we will be. The SE tells us the likely size of the random / chance error.
- SE $= \frac{\sigma}{\sqrt{n}}$, where σ is the standard deviation in the population. Suppose the standard deviation in the sample is 0.49, then $\frac{0.49}{\sqrt{1000}} \approx 1.6\%$.

The bootstrap

- What about statistics that are more complicated than an average? How do we estimate their SE?
- An easy and extremely popular solution is the **bootstrap**.

The bootstrap

- First insight: If we knew the sampling distribution, then we could find the SE because it is simply the standard deviation of the sampling distribution.

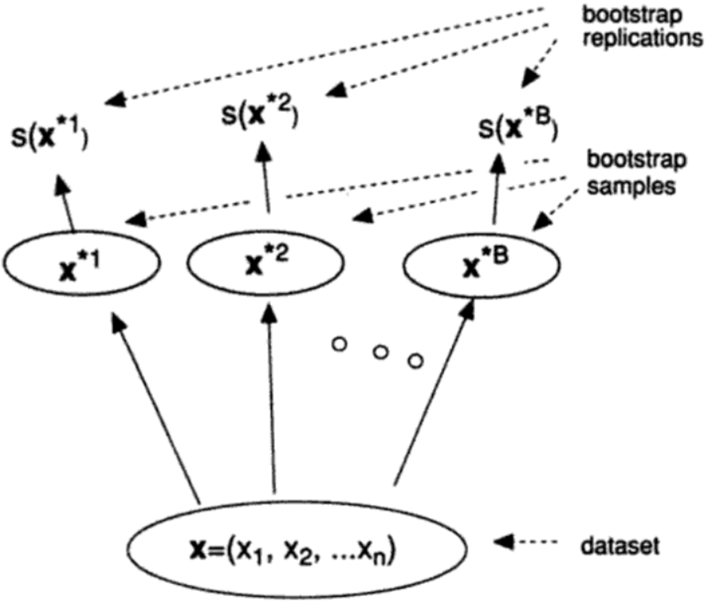


- But we don't know the sampling distribution, because we have only one estimate, not 10,000.

The bootstrap

- Second insight: We can create 10,000 “pseudo-samples” of size 50 by **sampling with replacement** from the data $x_1, ..., x_{50}$.
- Each “pseudo-sample” will give one “pseudo-estimate”.
- Make a histogram of these 10,000 pseudo-estimates and pretend that this is the sampling distribution.
- That’s how the bootstrap works.

Bootstrap


bootstrap replications
s(x*1)
s(x*2)
s(x*B)
bootstrap samples
x*1
x*2
x*B
dataset
x=(x1, x2, ...xn)

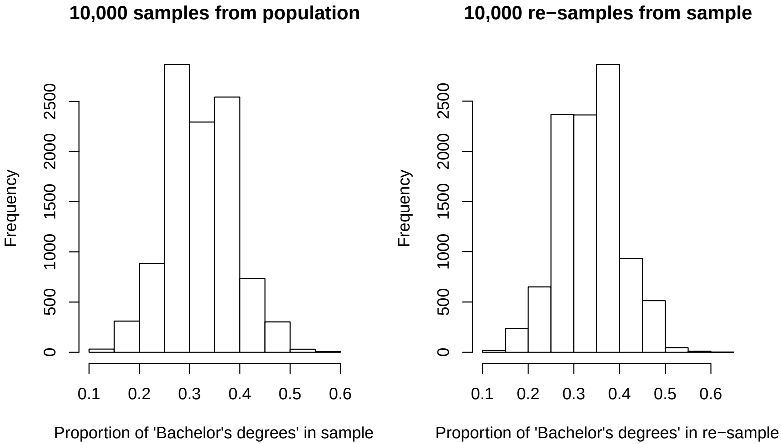
Population	Sample	Bootstrap samples
We do not observe	We obtain **one** sample from the population	We can resample with replacement from the sample **as many times as we want**

Population Statistic	Sample Statistic	Bootstrap Summaries
This is the **quantity we would like to know**, but we cannot observe	This is our **estimate** for the population summary	By calculating the standard deviation of the estimates from the different bootstrap samples we compute the **bootstrap SE** of our estimate

Why does the bootstrap work?

- It sounds crazy that sampling n times *with replacement* from the original data $x_1, ..., x_n$ (“re-sampling”) will produce a sample that is like a new sample from the population.
- The reason why this works *for producing a good approximation to the sampling distribution of an estimate* is because the histogram of the data $x_1, ..., x_n$ is close to the histogram of the population. We are able to replicate the chance process of drawing from an unknown *population* by drawing with replacement from our observed sample.
- It “works” in the sense that it gives a pretty good estimate of the sampling variability, *even when we do not know the exact population generating our sample.* (And we don't have to use any complicated calculations to get this estimate.)

The bootstrap



The bootstrap approximates the unknown sampling distribution (left histogram) by the *bootstrapped sampling distribution* (right histogram).

Example: Small population

Here is a small **population with its true mean**:

```r
population <- c(3,3,3,5,5,5,7,7,7,9,9,9,10,10,10)
mean(population)
```

```
[1] 6.8
```

Here is our **sample with its sample mean**:

```r
actualsample <- sample(population,5,replace=F)
actualsample
```

```
[1]  9  3 10  5  7
```

```r
mean(actualsample)
```

```
[1] 6.8
```

Example: bootstrap samples

Now we construct multiple bootstrap samples, sampling with replacement from:

```
[1]  9  3 10  5  7
```

The first 10 bootstrap samples (out of 10,000):

```
             B.obs1 B.obs2 B.obs3 B.obs4 B.obs5 B.Mean
boot.sample1      9      5      7      5      7    6.6
boot.sample2      9      9      9      3      3    6.6
boot.sample3      5      5      7      7      7    6.2
boot.sample4      7     10      9      3      5    6.8
boot.sample5      5     10      3      9      3    6.0
boot.sample6      5      9      9     10      5    7.6
boot.sample7      9      5      9      9      9    8.2
boot.sample8      3     10     10     10     10    8.6
boot.sample9      7      5      7      3      3    5.0
boot.sample10     3      7      3      7      5    5.0
```

Code for previous slide

```r
B <- 10000
bsample<-matrix(NA,B,5)
bsamplemeans <- matrix(nrow=B)

for(i in 1:B){
  bootsample <- sample(actualsample,5,replace=T)
  bsample[i,] <- bootsample
  bsamplemeans[i] <- mean(bootsample)
}

boot <- cbind(bsample, bsamplemeans)
rownames(boot) <- paste0("boot.sample", 1:B)
colnames(boot) <- c("B.obs1","B.obs2","B.obs3","B.obs4","B.obs5","B.Mean")
```

A toy example, evaluating variability

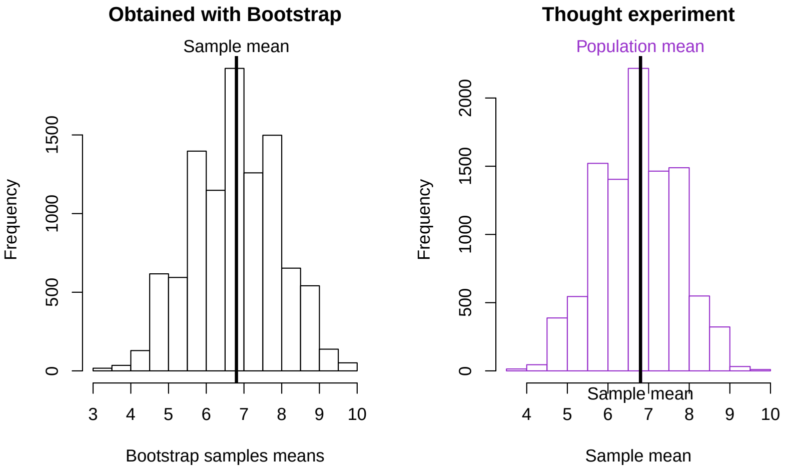
We can calculate the standard deviation of the bootstrap sample to estimate the SE of the sample estimate:

```
sqrt(var(bsamplemeans))
```

```
         [,1]
[1,] 1.153584
```

A toy example, checking that the bootstrap helped

Histograms of variability of sample estimate:



Does the bootstrap always work?

- If our sampling process is nothing like the bootstrap, then drawing repeatedly from our observed sample will not help us.
- For example, we re-sampled observations **uniformly at random** from our sample, and there was no **dependence** between successive samples.
- **We have to believe that the resampling process we use is a reasonable approximation to the way our data was sampled**.
- We might call this a **no free lunch principle**: We can use the computer to do calculations for us about the sampling processes **only** when we know how to tell the computer to mimic the sampling process.
- Put another way, computers **cannot magically create information where none exists**.

Recap

- We have looked at ways to estimate the SE.
- The SE tells us the likely size of the chance error.
- But confidence intervals can give us a more precise statement.
- What are confidence intervals?

Confidence Intervals (CI) and CLT for average

- According to the central limit theorem, the sample average follows a Normal distribution with expected value $\mu = 60\%$ and standard deviation $\frac{\sigma}{\sqrt{n}}$ (for large n).

In a Normal distribution:

- about 2/3 (68%) of the data fall within one standard deviation of the mean
- about 95% fall within 2 standard deviations of the mean
- about 99.7% fall within 3 standard deviations of the mean

Confidence Intervals (CI) and CLT

- Let's focus on what is called a "95% Confidence Interval".
- There is a 95% chance that the sample average is no more than 2 standard deviations away from the population parameter.
- But this is the same as saying that the population parameter is no more than 2 standard deviations away from the sample estimate.
- A confidence interval gives a range of plausible values for a population parameter μ.

Example Continued

- Recall that in the example about the US President's approval the sample approval percentage was 58% and that $\frac{0.49}{\sqrt{1000}} \approx 1.6\%$.
- Then [54.8%, 61.2%] is called a 95% **confidence interval** for the population percentage.
- Why 'confidence' and not 'probability'? The population parameter μ is a fixed number, which either falls into [54.8%, 61.2%] or not. There are no chances involved.
- Rather, the chances are in the sampling procedure: A different sample of 1,000 voters will give a slightly different interval.

Example continued

- If one does many polls, then 95% of these intervals trap the population percentage, and 5% will miss it.
- "I am 95% confident that the President's approval rating is between 54.8% and 61.2%" means that 95% of the time I am correct when making such a statement based on a poll.

Keep in mind that the interval varies from sample to sample, while the population percentage is a fixed number.

How we have calculated CI so far

- To calculate a confidence interval, so far we have leveraged the central limit theorem which works for averages (or sums).
- Then our confidence intervals had a simple form: $\bar{x} \pm 2 \times \sigma/\sqrt{n}$ (from a Normal distribution approximation) - this applies if the estimate is an average. Can estimate σ from the sample.
- But what if we want to calculate the CI for something more complicated? For example, what if we don't have an average and we don't know a formula for the SE? Or what if we don't have a Normal distribution?

Using the bootstrap to calcualte CI

- The bootstrap is more general and also simpler, but it can be computationally intensive.
- Using the bootstrap we can:
  1. (i) Use the bootstrap estimated SE, so the CI will be sample statistic $\pm 2 \times$ (SE) (Normal approximation).
  2. (ii) Find the "middle" 95% bootstrap statistics (i.e. look at the 2.5 and 97.5 percentiles).

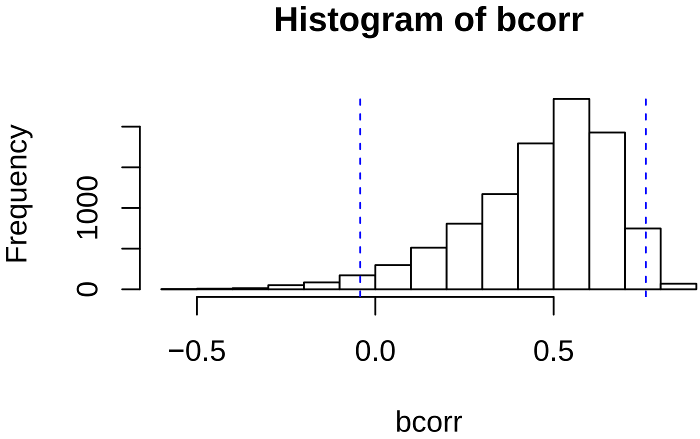
Example: Bootstrap CI for Claridge data (genetics and lefthandedness)

```r
library(boot)
data("claridge")

set.seed(44)
B<-10000

ssize<-nrow(claridge)
bcorr<-NULL

for(i in 1:B) {
  bsample<-claridge[sample(1:ssize,ssize,replace=TRUE),]
  bcorr<-c(bcorr,cor(bsample[,1],bsample[,2]))
}

qq=quantile(bcorr,c(.025,.975))
qq
```

```
       2.5%       97.5%
-0.04233162  0.75848145
```

Histogram of Bootstrap Correlations (with CI)

```r
hist(bcorr, col = "white")
abline(v=qq[1],lty=2, col = "blue")
abline(v=qq[2],lty=2, col = "blue")
```


Histogram of bcorr
Frequency
bcorr