Samping and bootstrapping

Seung-Ho An, University of Arizona

Agenda

Sampling framework

Polls

Random variables and probability distributions

Sampling distribution

Normal variables and the Central Limit Theorem

Resampling from our sample

Confidence intervals

Calculating confidence intervals

Sampling framework

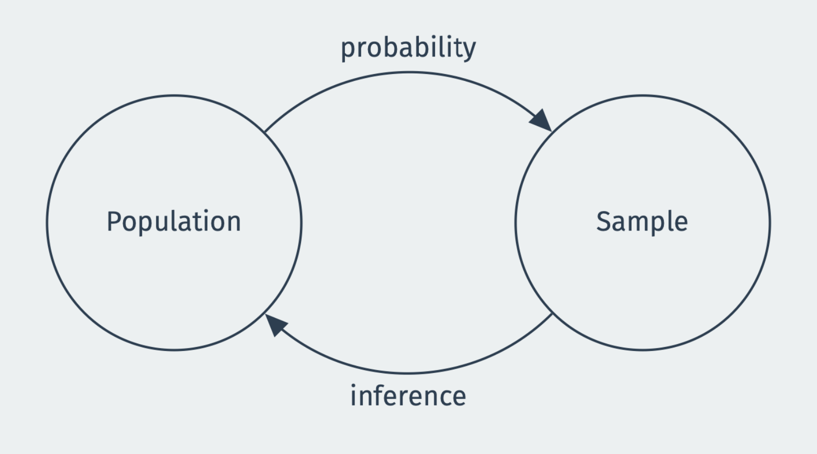
Population: group of units/people we want to learn about

Population parameter: some numerical summary of the population we would like to know - population mean/proportion, population standard deviation

Census: complete recording of data on the entire population

Sample: subset of the population taken in some way (hopefully randomly)

Estimator or sample statistic: numerical summary of the sample that is our "best guess" for the unknown population parameter


probability
Population
Sample
inference

Random sample: units selected into sample from population with a non-zero probability

Simple random sample: all units have the same probability of being selected into the sample

The **expected value** of a sample statistic, $\mathbb{E}[\hat{p}]$, is the average value of the statistic across repeated samples

The **expected value** of a sample proportion from a simple random sample is equal to the population proportion, $\mathbb{E}[\hat{p}] = p$

The **standard error** is the standard deviation of the sample statistic across repeated samples

Tells us how far away, on average, the sample proportion will be from the population proportion

Standard error vs. population standard deviation

The **standard error** is the SD of the statistic across repeated samples

Should not be confused with the population standard deviation or sample standard deviation, both of which measure how far **units** are away from a mean

Polls

- What proportion of the public approves of Biden's job as president?
- Gallup poll results from Sept 1st to 16th:
  - 812 adult Americans
  - Telephone interviews
  - Approve (42%), Disapprove (56%)

- **Population**: adults 18+ living in 50 US states and DC
- **Population parameter**: population proportion of all US adults that approve of Biden
  - Census: not possible
- **Sample**: random digit dialing phone numbers (cell and landline)
- **Point estimate** : sample proportion that approve of Biden

Random variables and probability distributions

Random variables are numerical summaries of chance processes:

$$X_i = \begin{cases} 1, & \text{if respondent } i \text{ supports Biden,} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

With a simple random sample, chance of $X_i = 1$ is equal to the population proportion of people that support Biden

- **Discrete**: X can take a finite (or countably infinite) number of values
  - Number of heads in 5 coin flips
  - Sampled senator is a woman (X=1) or not (X=0)
  - Number of battle deaths in a civil war
- **Continuous**: X can take any real value (usually within an interval)
  - GDP per capita (average income) in a county
  - Share of population that approves of Biden
  - Amount of time spent on a website

Probability distributions tell us the chances of different values of a r.v. occurring

Discrete variables: like a frequency barplot for the population distribution

Continuous variables: like a continuous version of population histogram

Sampling distribution

Suppose $X_1, X_2, ..., X_n$ is a simple random sample from a population distribution with mean μ ("mu") and variance σ^2 ("sigma squared")

Sample mean: $\overline{X}_n = \frac{1}{n}\Sigma_{i=1}^{n} X_i$

$$\overline{X}_n = \frac{X_1 + X_2 + ... + X_n}{n}$$

...

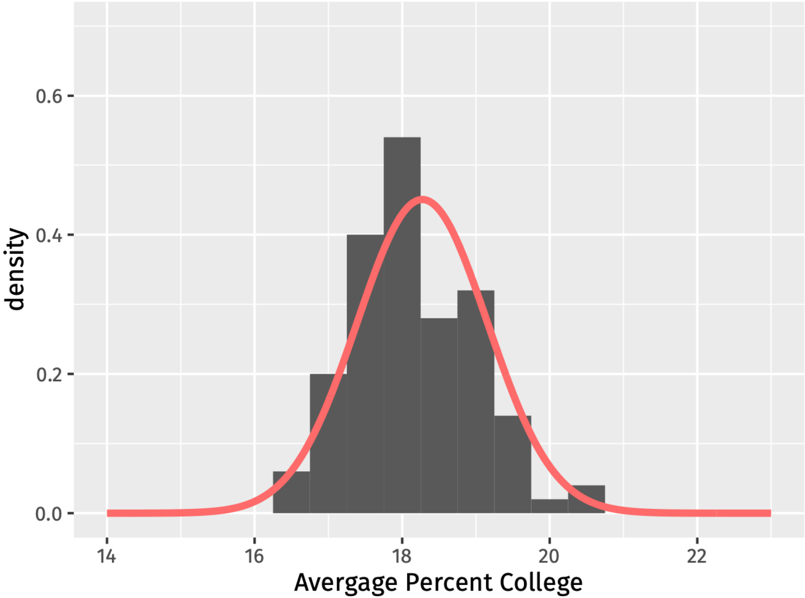
$\overline{X}_n$ is a random variable with a distribution!!

Sampling distributions are the probability distributions of an estimator like $\overline{X}_n$

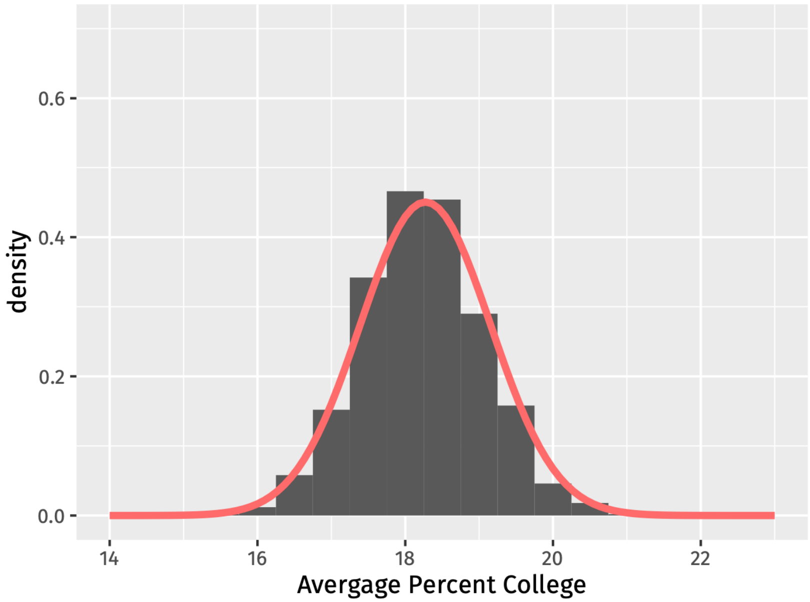
When we have access to the full population, we can approximate the sampling distribution with repeated sampling

Let's sample 50 observations from a sample distribution and repeat the process 100, 1,000, 10,000, and 100,000 times!

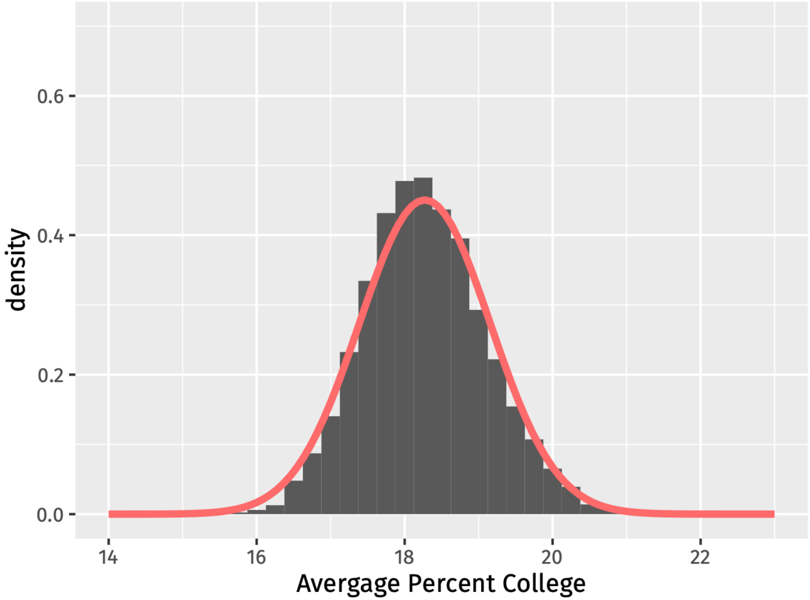
100 Repititions



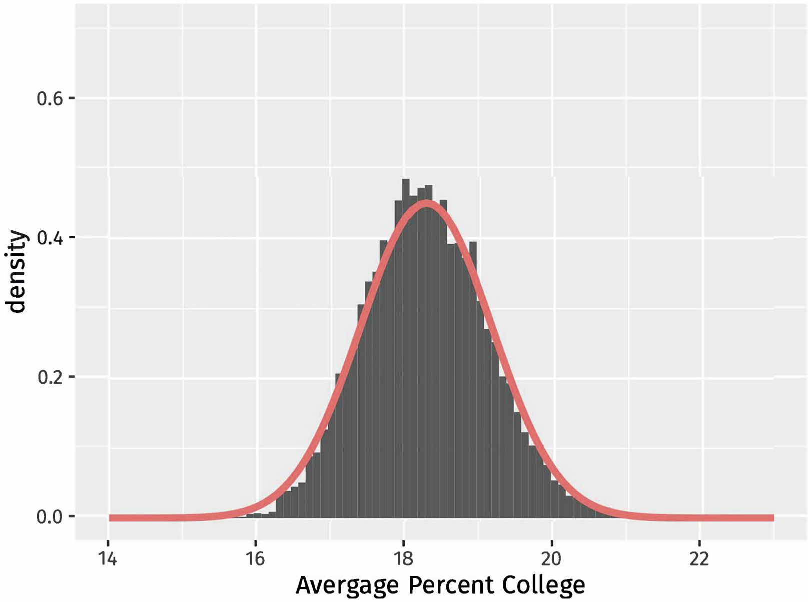
1,000 Repititions



10,000 Repititions



100,000 Repititions



Sampling distribution of the sample mean

Suppose $X_1, X_2, ..., X_n$ is a simple random sample from a population distribution with mean μ and variance σ^2

Expected value of the distribution of $\overline{X}_n$ is the population mean, μ

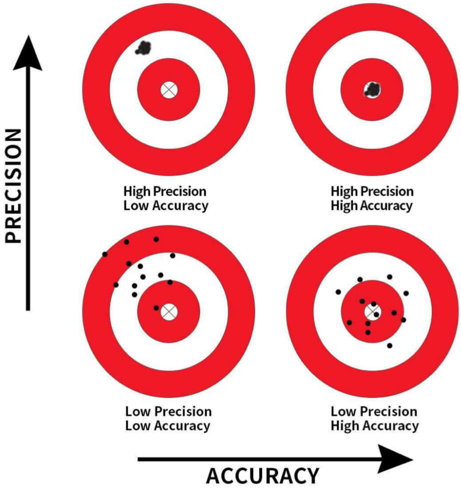
Standard error of the distribution of $\overline{X}_n$ is approximately $\sigma/\sqrt{n}$:

$$SE \approx \frac{\text{population standard deviation}}{\sqrt{\text{sample size}}}$$

An estimator is **unbiased** when its expected value across repeated samples equals the population parameter of interest

Sample mean of a simple random sample is **unbiased** for the population mean, $\mathbb{E}[\overline{X}_n] = \mu$

An estimator that isn't unbiased is called **biased**


PRECISION
High Precision
Low Accuracy
High Precision
High Accuracy
Low Precision
Low Accuracy
Low Precision
High Accuracy
ACCURACY

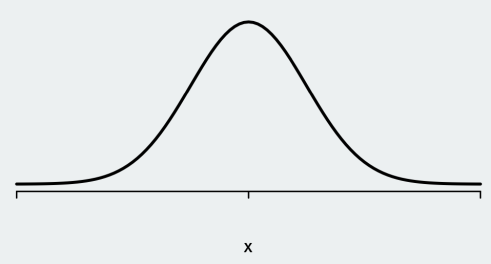
Law of large numbers

Let $X_1, ..., X_n$ be a simple random sample from a population with mean μ and finite variance σ^2. Then, $\overline{X}_n$ converges to μ as *n* gets large

- Probability of $\overline{X}_n$ being "far away" from μ goes to 0 as n gets big
- The distribution of sample mean "collapses" to population mean
- Can see this from the SE of $\overline{X}_n$: $SE = \sigma/\sqrt{n}$
- Not necessarily true with a biased sample

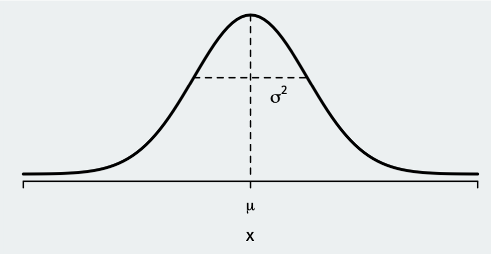
Normal variables and the Central Limit Theorem

Normal random variable



- A **normal distribution** has a PDF (probability density function) that is the classic "bell-shaped" curve
  - Extremely ubiquitous in statistics
  - An r.v. is more likely to be in the center, rather than the tails
- Three key properties of this PDF:
  - **Unimodal**: one peak at the mean
  - Symmetric around the mean
  - Everywhere positive: the function always has values greater than, or equal to, 0

Normal distribution

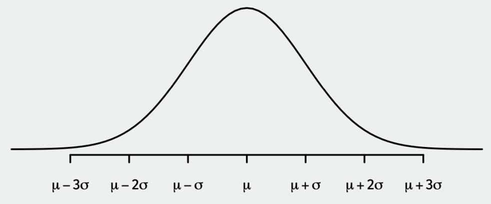


- A normal distribution can be affected by two values:
  - **mean/expected value** usually written as μ
  - **variance** written as σ^2 (standard deviation is σ)
  - Written X $N(\mu, \sigma^2)$
- **Standard normal distribution**: mean 0 and standard deviation 1

Central limit theorem

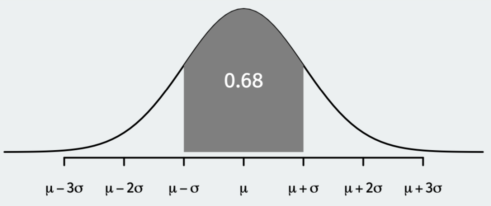
Let $X_1, \ldots, X_n$ be a simple random sample from a population with mean μ and finite variance σ^2. Then, $\overline{X}_n$ will be approximately distributed $N(\mu, \sigma^2/n)$ in large samples

- "Sample means tend to be normally distributed as samples get large"
- $\rightsquigarrow$ we know (an approx. of) the entire probability distribution of $\overline{X}_n$
  - Approximation is better as n goes up
  - Does not depend on the distribution of X_i



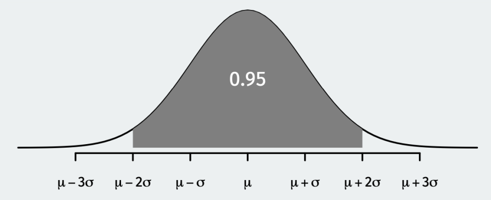
- If X $N(\mu, \sigma^2)$, then:

Empirical rule for the normal distribution



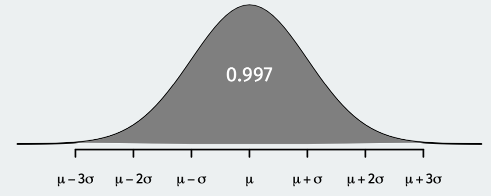
- If X $N(\mu, \sigma^2)$, then:
  - $\approx$ 68% of the distribution of X is within 1 SD of the mean

Empirical rule for the normal distribution

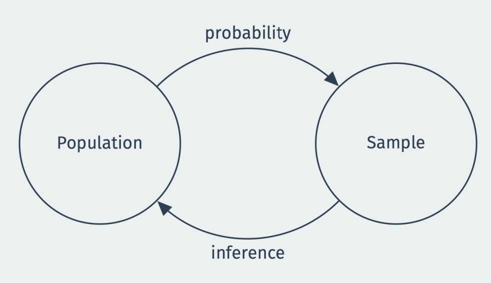


- If X $N(\mu, \sigma^2)$, then:
  - $\approx$ 68% of the distribution of X is within 1 SD of the mean
  - $\approx$ 95% of the distribution of X is within 2 SD of the mean

Empirical rule for the normal distribution



- If X $N(\mu, \sigma^2)$, then:
  - $\approx$ 68% of the distribution of X is within 1 SD of the mean
  - $\approx$ 95% of the distribution of X is within 2 SDs of the mean
  - $\approx$ 99.7% of the distribution of X is within 3 SDs of the mean
- CLT + empirical rule: we'll know the rough distribution of estimation errors we should expect



We only get 1 sample. Can we learn about the population from that sample?

Resampling from our sample (bootstrapping)

Variable	Description
state	State of respondent
district	Congressional district of respondent
pid7	Party ID (1=Strong D, 7=Strong R)
pres_vote	Self reported vote in 2020
sci_therm	0-100 therm score for scientists
rural_therm	0-100 therm score for rural Americans
favor_voter_id	1 if respondent thinks voter ID should be required
envir_doing_more	1 if respondent thinks gov't should be doing more climate change

```r
library(TPDdata)
anes
```

```
## # A tibble: 5,162 x 8
##    state district  pid7 pres_vote sci_therm rural_therm favor_voter_id envir_d~1
##    <chr>    <dbl> <dbl> <chr>         <dbl>       <dbl>          <dbl>     <dbl>
##  1 ID           2     4 Other            70          60              1         0
##  2 VA           2     3 Biden           100          75              0         1
##  3 CO           4     4 Trump            60          90              1         0
##  4 TX           5     3 Biden            85          85              1         1
##  5 WI           6     6 Trump            85          70              1         1
##  6 CA          40     2 Biden            50          50              1         0
##  7 WI           5     2 Biden           100          70              1         0
##  8 OR           4     7 Trump            70          50              0         1
##  9 MA           5     3 Biden            80          70              0         1
## 10 NV           3     1 Biden            85          40              0         1
## # ... with 5,152 more rows, and abbreviated variable name 1: envir_doing_more
```

What is the average thermometer score for scientists?

```
anes |>
  summarize(mean(sci_therm))
```

```
## # A tibble: 1 x 1
##   `mean(sci_therm)`
##               <dbl>
## 1              80.6
```

What is the sampling distribution of this average? We only have this 1 draw!

Population: all US adults

Population parameter: average feeling thermometer score for scientists among all US adults

Sample: (complicated) random sample of all US adults

Sample statistic/point estimate: sample average of thermometer scroes

Roughly how far our point estimate is likely to be from the truth?

Mimic sampling from the population by **resampling** many times from the sample itself

Bootstrap resampling done **with replacement** (same row can appear more than once)

One bootstrap resample

```r
boot_1 <- anes |>
  slice_sample(prop = 1, replace = TRUE)
boot_1
```

```
## # A tibble: 5,162 x 8
##    state district  pid7 pres_vote sci_therm rural_therm favor_voter_id envir_d~1
##    <chr>    <dbl> <dbl> <chr>         <dbl>       <dbl>          <dbl>     <dbl>
##  1 MD           8     4 Biden            70          60              0         0
##  2 CO           7     3 Biden           100          45              1         1
##  3 UT           2     2 Biden           100          50              0         1
##  4 CA          12     1 Biden           100         100              0         1
##  5 NH           1     4 Biden            85          60              1         1
##  6 WI           4     1 Biden           100          98              1         1
##  7 AZ           8     6 Trump            85         100              1         0
##  8 VA           8     1 Biden           100          60              0         1
##  9 WA           8     3 Biden            70          50              0         1
## 10 DC           1     1 Biden            85          30              0         1
## # ... with 5,152 more rows, and abbreviated variable name 1: envir_doing_more
```

Sample mean in the bootstrap sample

```
boot_1 |>
  summarize(mean(sci_therm))
```

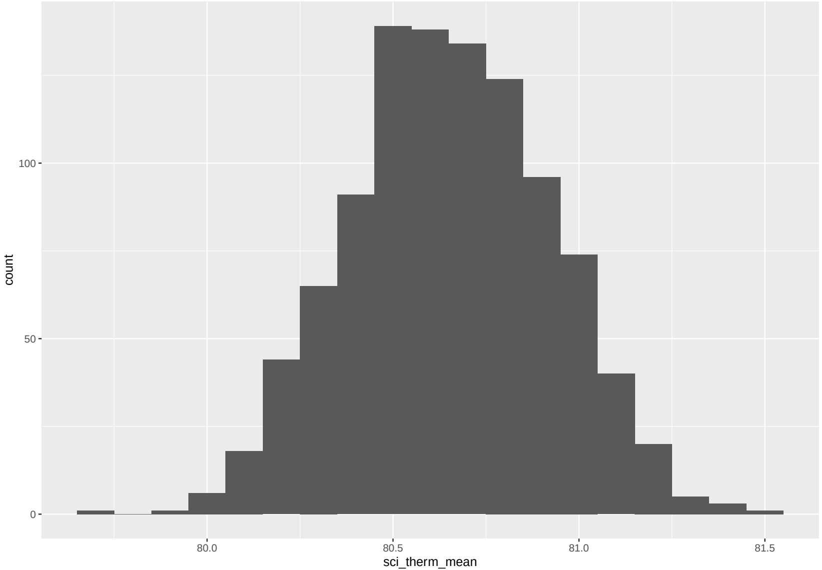
```
## # A tibble: 1 x 1
##   `mean(sci_therm)`
##               <dbl>
## 1              81.2
```

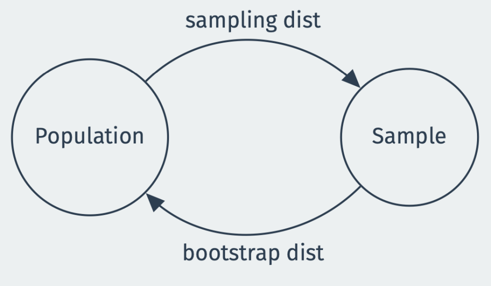
```r
library(infer)
bootstrap_dist<-anes |>
  rep_sclice_sample(prop=1, reps=1000, replace=TRUE) |>
  group_by(replicate) |>
  summarize(sci_therm_mean = mean(sci_therm))
bootstrap_dist
```

```
## # A tibble: 1,000 x 2
##    replicate sci_therm_mean
##        <int>          <dbl>
##  1         1           80.4
##  2         2           80.5
##  3         3           80.5
##  4         4           80.1
##  5         5           81.1
##  6         6           80.7
##  7         7           80.7
##  8         8           81.1
##  9         9           81.1
## 10        10           80.7
## # ... with 990 more rows
```

Visualizing the bootstrap distribution

```
bootstrap_dist |>
  ggplot(aes(x = sci_therm_mean)) +
  geom_histogram(binwidth=0.1)
```





Bootstrap distribution **approximates** the sampling distribution of the estimator

Both should have a **similar shape and spread** if sampling from the distribution $\approx$ bootstrap resampling

Approximation gets better as sample gets bigger

Given the sampling, not surprising that bootstrap distribution is centered on the point estimate:

```
bootstrap_dist |>
  summarize(mean(sci_therm_mean))
```

```
## # A tibble: 1 x 1
##   `mean(sci_therm_mean)`
##                    <dbl>
## 1                   80.7
```

```
anes |>
  summarize(mean(sci_therm))
```

```
## # A tibble: 1 x 1
##   `mean(sci_therm)`
##               <dbl>
## 1              80.6
```

Confidence intervals

What is a confidence interval?

Point estimate: best single guess about the population parameter. Unlikely to be exactly correct

Confidence interval: a range of plausible values of the population parameter

- Each sample gives a different CI or toss of the ring
- Some samples the ring will will contain the target (the CI will contain the truth) other times it won't
  - We don't know if the CI for our sample contains the truth
- **Confidence level**: percent of the time our CI will contain the population parameter
  - Number of ring tosses that will hit the target
  - We get to choose, but typical values are 90%, 95%, and 99%

Confidence intervals as occational liars

The **confidence level** of a CI determines how often the CI will be wrong

A 95% confidence interval will:

- Tell you the truth in 95% of repeated samples (contain the population parameter 95% of the time)
- Lie to you in 5% of repeated sample (not contain the population parameter 5% of the time)

Can you tell if your particular confidence interval is telling the truth?
No!

Calculating confidence intervals

Possible to use `quantile` to calculate CIs, but `infer` package is a more unified framework for CIs and hypothesis tests

We'll use a dplyr like approach of chained calls

Step 1: define an outcome of interest

Start with defining the variable of interest:

```
anes |>
  specify(response = sci_therm)
```

```
## Response: sci_therm (numeric)
## # A tibble: 5,162 x 1
##    sci_therm
##        <dbl>
##  1        70
##  2       100
##  3        60
##  4        85
##  5        85
##  6        50
##  7       100
##  8        70
##  9        80
## 10        85
## # ... with 5,152 more rows
```

Step 2: generate bootstraps

Next `infer` can generate bootstraps with the `generate()` function (similar to `rep_slice_sample()`):

```
anes |>
  specify(response = sci_therm) |>
  generate(reps = 1000, type="bootstrap")
```

```r
anes |>
  specify(response = sci_therm) |>
  generate(reps = 1000, type="bootstrap")
```

```
## Response: sci_therm (numeric)
## # A tibble: 5,162,000 x 2
## # Groups:   replicate [1,000]
##    replicate sci_therm
##        <int>     <dbl>
##  1         1        95
##  2         1        60
##  3         1       100
##  4         1        70
##  5         1        60
##  6         1        85
##  7         1        85
##  8         1        85
##  9         1       100
## 10         1        85
## # ... with 5,161,990 more rows
```

Use `calculate()` to do the group_by(`replicate`) and `summarize` commands in one:

```
boot_dist_infer <- anes |>
  specify(response = sci_therm) |>
  generate(reps = 1000, type = "bootstrap") |>
  calculate(stat = "mean")
```

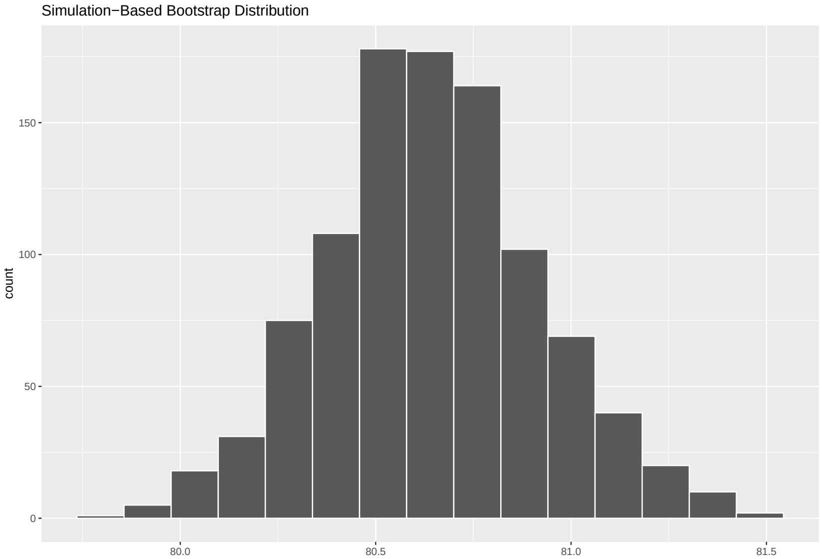
```
boot_dist_infer
```

```
## Response: sci_therm (numeric)
## # A tibble: 1,000 x 2
##    replicate  stat
##        <int> <dbl>
##  1         1  80.6
##  2         2  80.5
##  3         3  80.7
##  4         4  80.9
##  5         5  80.4
##  6         6  80.4
##  7         7  80.5
##  8         8  80.5
##  9         9  80.6
## 10        10  80.6
## # ... with 990 more rows
```

Step 3(b) visualize the bootstrap distribution

`infer` also has a shortcut for plotting called `visualize()`

```
visualize(boot_dist_infer)
```



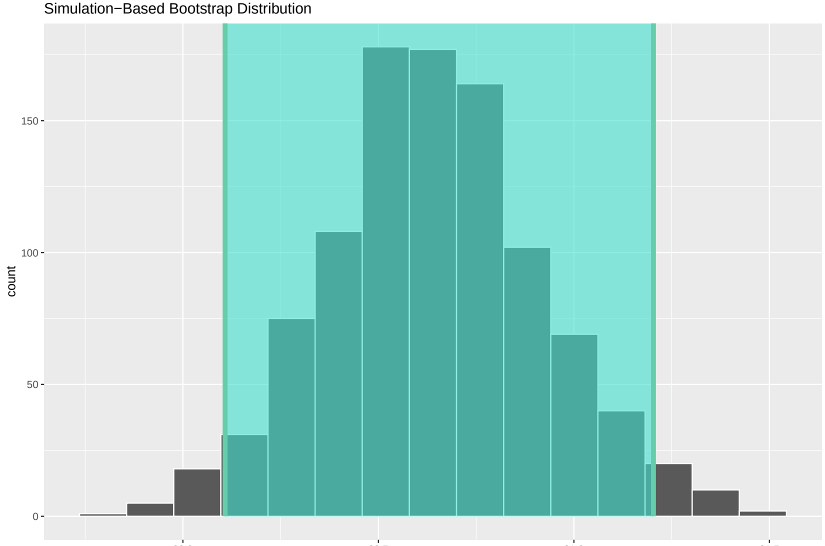
Finally we can calculate the CI using the percentile method with `get_confidence_interval()`:

```
perc_ci_95 <- boot_dist_infer |>
  get_confidence_interval(level = 0.95, type = "percentile")
perc_ci_95
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     80.1     81.2
```

Step 4(b): visualize CIs

```
visualize(boot_dist_infer) +
  shade_confidence_interval(endpoints = perc_ci_95)
```

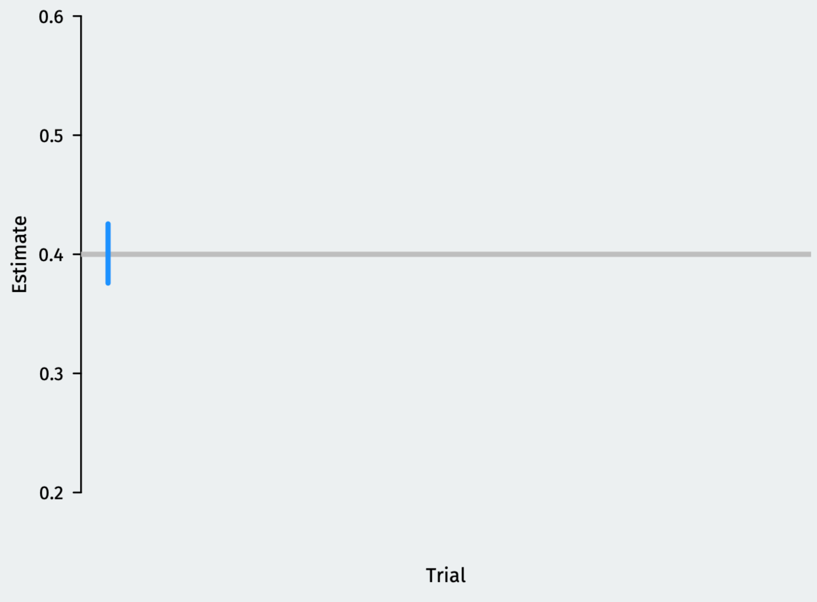


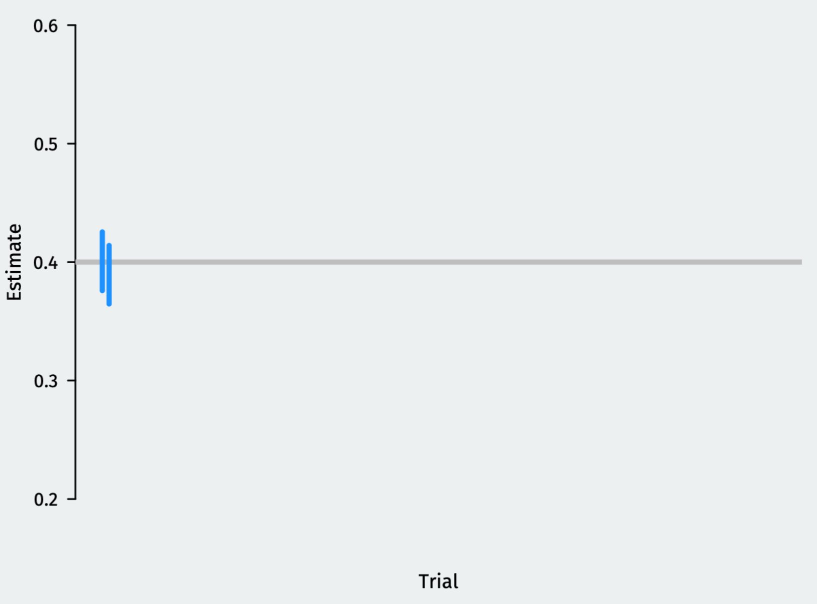
Interpreting confidence intervals

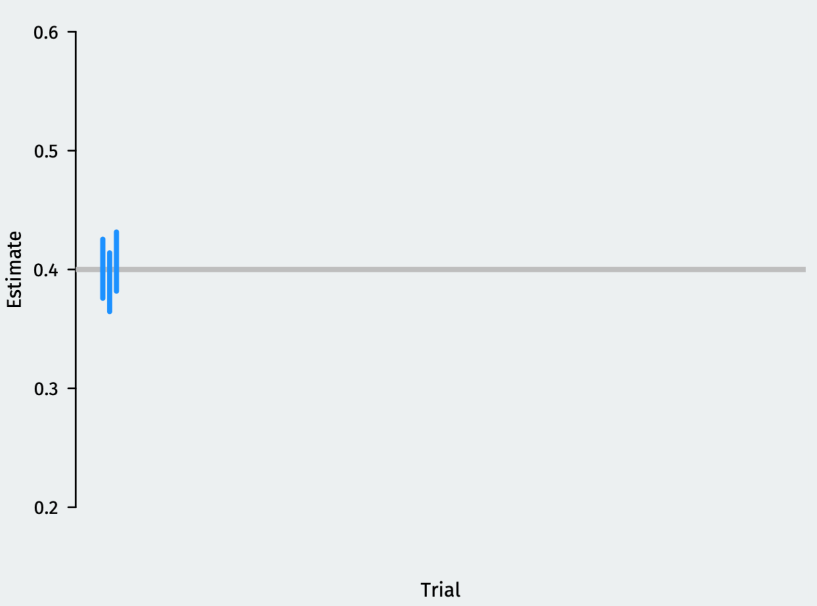
- Be careful about interpretation:
  - A 95% confidence interval will contain the true value in 95% of repeated samples
  - For a particular calculated confidence interval, truth is either in it or not
- A simulation can help our understanding:
  - Draw samples of size 1500 assuming population approval for Biden of p=0.4
  - Calculate 95% confidence intervals in each sample
  - See how many overlap with the true population approval

Plotting the CIs

Plotting the CIs




0.6
0.5
0.4
0.3
0.2
Estimate
Trial


0.6
0.5
0.4
0.3
0.2
Estimate
Trial

Plotting the CIs