

Lab5b_606 - Foundations for statistical inference - Confidence intervals

Heleine Fouda

2023-10-09

If you have access to data on an entire population, say the opinion of every adult in the United States on whether or not they think climate change is affecting their local community, it's straightforward to answer questions like, "What percent of US adults think climate change is affecting their local community?". Similarly, if you had demographic information on the population you could examine how, if at all, this opinion varies among young and old adults and adults with different leanings. If you have access to only a sample of the population, as is often the case, the task becomes more complicated. What is your best guess for this proportion if you only have data from a small sample of adults? This type of situation requires that you use your sample to make inference on what your population looks like.

Setting a seed: You will take random samples and build sampling distributions in this lab, which means you should set a seed on top of your lab. If this concept is new to you, review the lab on probability.

Getting Started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages, and perform statistical inference using **infer**.

Let's load the packages.

```
library(tidyverse)
library(openintro)
library(infer)
```

The data

A 2019 Pew Research report states the following:

To keep our computation simple, we will assume a total population size of 100,000 (even though that's smaller than the population size of all US adults).

Roughly six-in-ten U.S. adults (62%) say climate change is currently affecting their local community either a great deal or some, according to a new Pew Research Center survey.

Source: Most Americans say climate change impacts their community, but effects vary by region

In this lab, you will assume this 62% is a true population proportion and learn about how sample proportions can vary from sample to sample by taking smaller samples from the population. We will first create our population assuming a population size of 100,000. This means 62,000 (62%) of the adult population think climate change impacts their community, and the remaining 38,000 does not think so.

```
us_adults <- tibble(
  climate_change_affects = c(rep("Yes", 62000), rep("No", 38000))
)
```

The name of the data frame is `us_adults` and the name of the variable that contains responses to the question “Do you think climate change is affecting your local community?” is `climate_change_affects`.

We can quickly visualize the distribution of these responses using a bar plot.

```
ggplot(us_adults, aes(x = climate_change_affects)) +  
  geom_bar() +  
  labs(  
    x = "", y = "",  
    title = "Do you think climate change is affecting your local community?"  
  ) +  
  coord_flip()
```



We can also obtain summary statistics to confirm we constructed the data frame correctly.

```
us_adults %>%  
  count(climate_change_affects) %>%  
  mutate(p = n / sum(n))  
  
## # A tibble: 2 x 3  
##   climate_change_affects    n    p  
##   <chr>                <int> <dbl>  
## 1 No                   38000 0.38  
## 2 Yes                  62000 0.62
```

In this lab, you’ll start with a simple random sample of size 60 from the population.

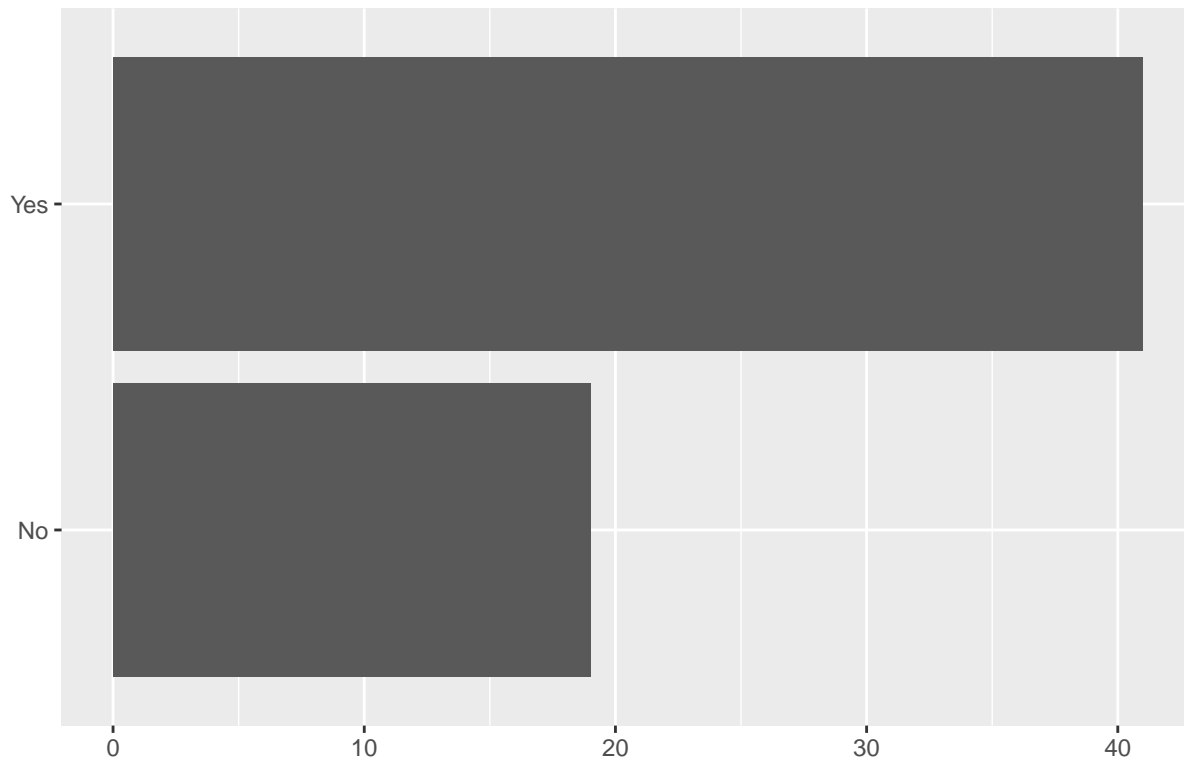
```
n <- 60  
samp <- us_adults %>%  
  sample_n(size = n)
```

1. What percent of the adults in your sample think climate change affects their local community? **Hint:** Just like we did with the population, we can calculate the proportion of those **in this sample** who think climate change affects their local community.

62% of the adults in the sample think climate change affects their lives.

```
# Barplot - samp60  
ggplot(samp, aes(x = climate_change_affects)) +  
  geom_bar() +  
  labs(  
    x = "", y = "",  
    title = "Do you think climate change is affecting your local community?"  
  ) +  
  coord_flip()
```

Do you think climate change is affecting your local community?



```
set.seed(131017)
# Summary statistics - samp60

n <- 60
samp <- us_adults %>%
  sample_n(size = n)

samp %>%
  count(climate_change_affects) %>%
  mutate(p60 = n / sum(n))
```

```
## # A tibble: 2 x 3
##   climate_change_affects    n  p60
##   <chr>                 <int> <dbl>
## 1 No                     19 0.317
## 2 Yes                     41 0.683
```

1. Would you expect another student's sample proportion to be identical to yours? Would you expect it to be similar? Why or why not?

The results would be different, yet similar. I would not expect another student's sample proportion to be identical to mine as this is random sampling. But, I would expect the proportions to be somehow similar based on the Central Limit Theorem and the effects of the law of large numbers

Confidence intervals

Return for a moment to the question that first motivated this lab: based on this sample, what can you infer about the population? With just one sample, the best estimate of the proportion of US adults who think

climate change affects their local community would be the sample proportion, usually denoted as \hat{p} (here we are calling it `p_hat`). That serves as a good **point estimate**, but it would be useful to also communicate how uncertain you are of that estimate. This uncertainty can be quantified using a **confidence interval**.

One way of calculating a confidence interval for a population proportion is based on the Central Limit Theorem,

Another way is using simulation, or to be more specific, using **bootstrapping**. The term **bootstrapping** comes from the phrase “pulling oneself up by one’s bootstraps”, which is a metaphor for accomplishing an impossible task without any outside help. In this case the impossible task is estimating a population parameter (the unknown population proportion), and we’ll accomplish it using data from only the given sample. Note that this notion of saying something about a population parameter using only information from an observed sample is the crux of statistical inference, it is not limited to bootstrapping.

In essence, bootstrapping assumes that there are more of observations in the populations like the ones in the observed sample. So we “reconstruct” the population by resampling from our sample, with replacement. The bootstrapping scheme is as follows:

- **Step 1.** Take a bootstrap sample - a random sample taken **with replacement** from the original sample, of the same size as the original sample.
- **Step 2.** Calculate the bootstrap statistic - a statistic such as mean, median, proportion, slope, etc. computed on the bootstrap samples.
- **Step 3.** Repeat steps (1) and (2) many times to create a bootstrap distribution - a distribution of bootstrap statistics.
- **Step 4.** Calculate the bounds of the XX% confidence interval as the middle XX% of the bootstrap distribution.

Instead of coding up each of these steps, we will construct confidence intervals using the **infer** package.

Below is an overview of the functions we will use to construct this confidence interval:

Function	Purpose
<code>specify</code>	Identify your variable of interest
<code>generate</code>	The number of samples you want to generate
<code>calculate</code>	The sample statistic you want to do inference with, or you can also think of this as the population parameter you want to do inference for
<code>get_ci</code>	Find the confidence interval

This code will find the 95 percent confidence interval for proportion of US adults who think climate change affects their local community.

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##   <dbl>     <dbl>
## 1     0.567     0.8
```

- In `specify` we specify the `response` variable and the level of that variable we are calling a `success`.

- In `generate` we provide the number of resamples we want from the population in the `reps` argument (this should be a reasonably large number) as well as the type of resampling we want to do, which is `"bootstrap"` in the case of constructing a confidence interval.
- Then, we `calculate` the sample statistic of interest for each of these resamples, which is `proportion`.

Feel free to test out the rest of the arguments for these functions, since these commands will be used together to calculate confidence intervals and solve inference problems for the rest of the semester. But we will also walk you through more examples in future chapters.

To recap: even though we don't know what the full population looks like, we're 95% confident that the true proportion of US adults who think climate change affects their local community is between the two bounds reported as result of this pipeline.

Confidence levels

1. In the interpretation above, we used the phrase "95% confident". What does "95% confidence" mean?

A 95% confidence interval means that there is 95% certainty our calculated range of values contains the population mean, leaving just 5% margin of error or uncertainty

In this case, you have the rare luxury of knowing the true population proportion (62%) since you have data on the entire population.

1. Does your confidence interval capture the true population proportion of US adults who think climate change affects their local community? If you are working on this lab in a classroom, does your neighbor's interval capture this value?

No, my confidence interval does exactly captures the true population proportion but it provides a range within which the true values can be found.

1. Each student should have gotten a slightly different confidence interval. What proportion of those intervals would you expect to capture the true population mean? Why?

Based on Figure 5.6 OpenIntro Statistics, 4th Edition (page 182). I would expect about 95% proportion of those intervals to capture the true population mean, because there is only one mean.

In the next part of the lab, you will collect many samples to learn more about how sample proportions and confidence intervals constructed based on those samples vary from one sample to another.

- Obtain a random sample.
- Calculate the sample proportion, and use these to calculate and store the lower and upper bounds of the confidence intervals.
- Repeat these steps 50 times.

Doing this would require learning programming concepts like iteration so that you can automate repeating running the code you've developed so far many times to obtain many (50) confidence intervals. In order to keep the programming simpler, we are providing the interactive app below that basically does this for you and created a plot similar to Figure 5.6 on OpenIntro Statistics, 4th Edition (page 182).

1. Given a sample size of 60, 1000 bootstrap samples for each interval, and 50 confidence intervals constructed (the default values for the above app), what proportion of your confidence intervals include the true population proportion? Is this proportion exactly equal to the confidence level? If not, explain why. Make sure to include your plot in your answer.

.633 to .733 intervals are included in the true population proportion, which is exactly equal to the confidence level. The proportions do not exactly equal the confidence intervals as proportions measure the true population value and intervals represent the range withing which these true values fall.

```
# Bootstrap
set.seed(131017)
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.50)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##   <dbl>    <dbl>
## 1     0.65     0.717
```

```
set.seed(131017)
n <- 60
samp <- us_adults %>%
  sample_n(size = n)

samp %>%
  count(climate_change_affects) %>%
  mutate(p60 = n / sum(n))
```

```
## # A tibble: 2 x 3
##   climate_change_affects    n  p60
##   <chr>                  <int> <dbl>
## 1 No                      19 0.317
## 2 Yes                     41 0.683
```

More Practice

1. Choose a different confidence level than 95%. Would you expect a confidence interval at this level to be wider or narrower than the confidence interval you calculated at the 95% confidence level? Explain your reasoning.

If I chose a confidence level greater than 95%, I would expect a narrower spread based on the premises of the normal distribution and the Central Limit Theorem

1. Using code from the **infer** package and data from the one sample you have (**samp**), find a confidence interval for the proportion of US Adults who think climate change is affecting their local community with a confidence level of your choosing (other than 95%) and interpret it.

Insert your answer here Lower CI = 0.607 Upper CI = 0.767

```
set.seed(131017)
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 60, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.85)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##   <dbl>    <dbl>
## 1     0.607     0.767
```

1. Using the app, calculate 50 confidence intervals at the confidence level you chose in the previous

question, and plot all intervals on one plot, and calculate the proportion of intervals that include the true population proportion. How does this percentage compare to the confidence level selected for the intervals?

Insert your answer here

This percentage calculated from the app is much higher than the selected confidence intervals.

```
# Let's build a function that creates a dataframe with a single record.
# that record will have three columns, sample_size, lower_ci and upper_ci.
# It's configured for bootstrap, 1,000 iterations and a CI of 95%.

calculate_ci_dataframe <- function(sample_size, ci_level) {
  results_df <- data.frame(sample_size = integer(0),
                           lower_ci = numeric(0), upper_ci = numeric(0),
                           ci_difference = numeric(0))

  result <- us_adults %>%
    sample_n(size = sample_size) %>%
    specify(response = climate_change_affects, success = "Yes") %>%
    generate(reps = 1000, type = "bootstrap") %>%
    calculate(stat = "prop") %>%
    get_ci(level = ci_level)

  ci_diff <- result$upper_ci - result$lower_ci

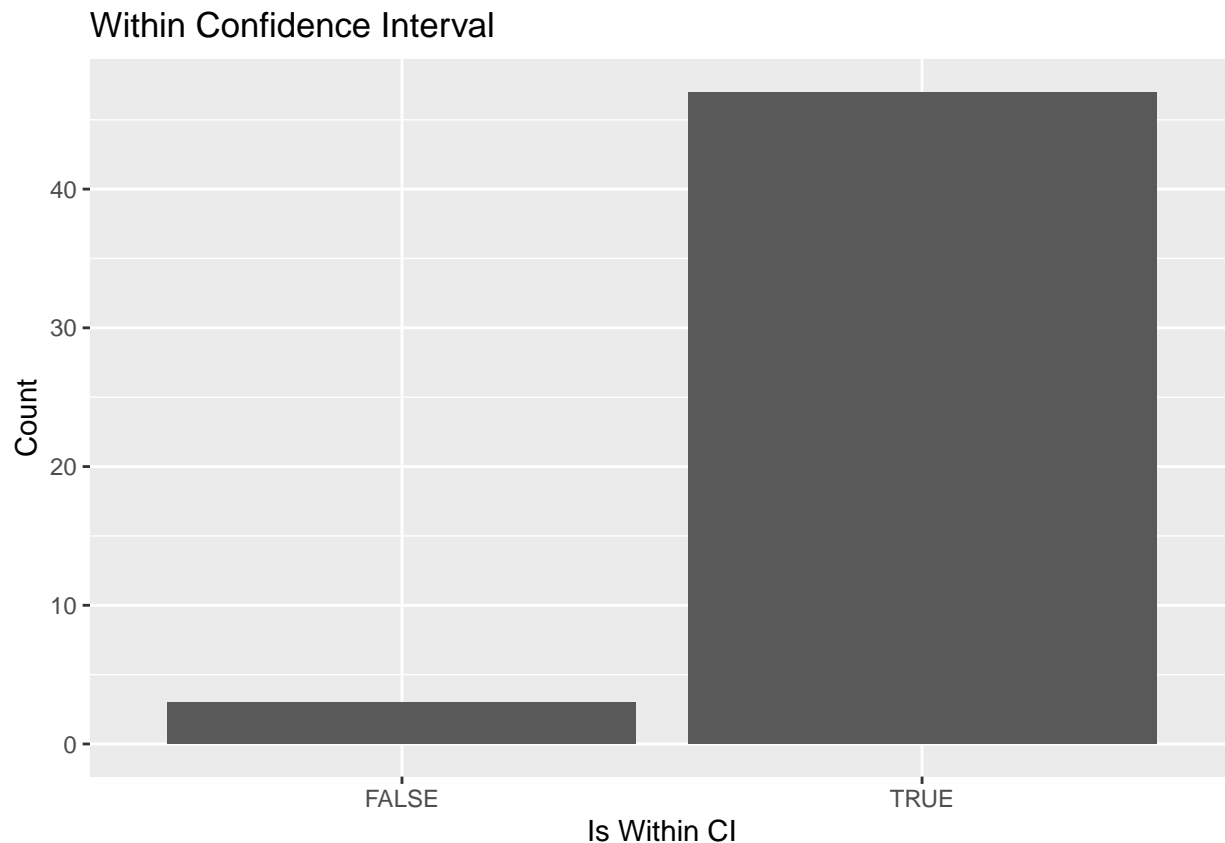
  results_df <- bind_rows(results_df,
                           data.frame(sample_size = sample_size,
                                       lower_ci = result$lower_ci,
                                       upper_ci = result$upper_ci,
                                       ci_difference = ci_diff))

  return(results_df)
}

# Now we run it 50 times, store it in a list.
results_list <- list()
for (i in 1:50) {
  result_df <- calculate_ci_dataframe(sample_size = 100, ci_level = 0.95)
  results_list[[i]] <- result_df
}

# now we'll convert it to a dataframe and then plot.
data95 <- do.call(rbind, results_list) %>%
  mutate(is_between = ifelse(0.62 >= lower_ci & 0.62 <= upper_ci, TRUE, FALSE))

ggplot(data95, aes(x = is_between)) +
  geom_bar() +
  labs(x = "Is Within CI", y = "Count", title = "Within Confidence Interval")
```



```
glimpse(data95)
```

```
## Rows: 50
## Columns: 5
## $ sample_size    <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 1~
## $ lower_ci       <dbl> 0.49000, 0.40000, 0.48000, 0.43975, 0.52000, 0.52000, 0.~
## $ upper_ci       <dbl> 0.69000, 0.60000, 0.67000, 0.63000, 0.72000, 0.72000, 0.~
## $ ci_difference   <dbl> 0.20000, 0.20000, 0.19000, 0.19025, 0.20000, 0.20000, 0.~
## $ is_between     <lgl> TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T~
```

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.607)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##   <dbl>    <dbl>
## 1    0.633    0.733
```

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.767)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
```

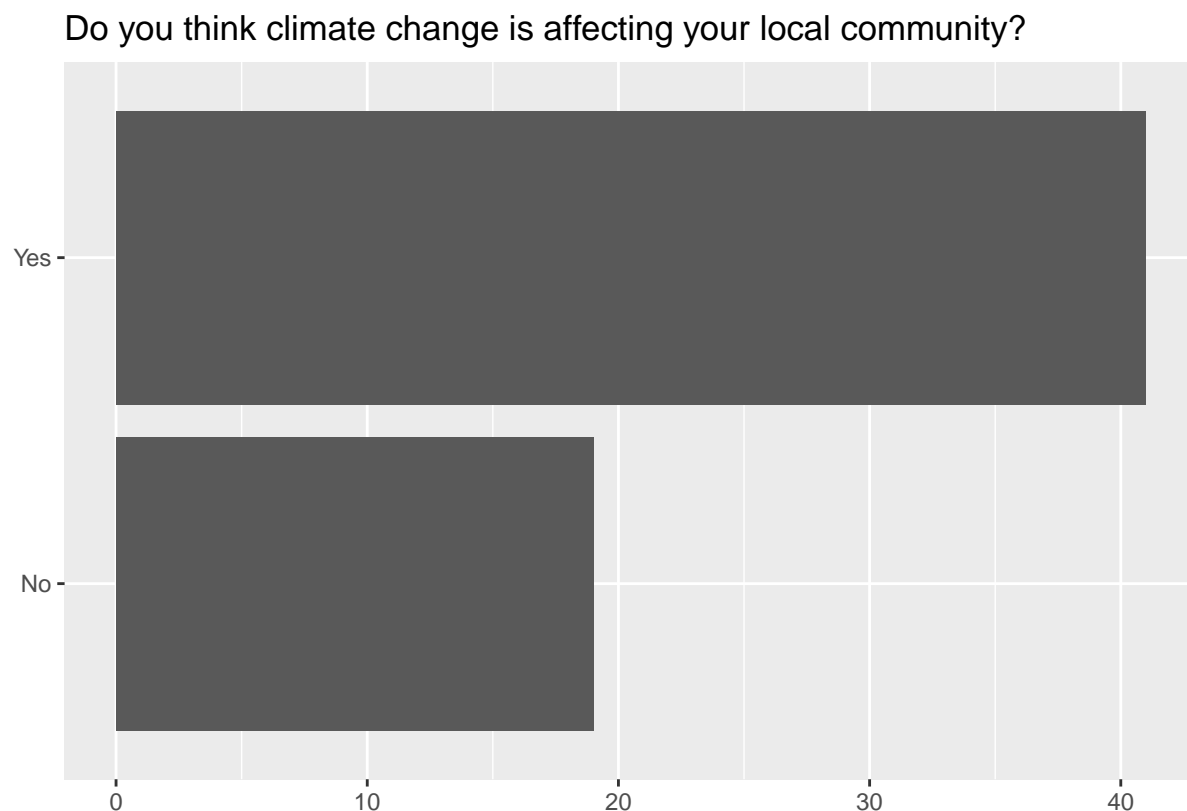


```
##      <dbl>      <dbl>
## 1      0.6      0.75

set.seed(131017)
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 60, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.85)

## # A tibble: 1 x 2
##   lower_ci upper_ci
##   <dbl>      <dbl>
## 1    0.607    0.767

ggplot(samp, aes(x = climate_change_affects)) +
  geom_bar() +
  labs(
    x = "", y = "",
    title = "Do you think climate change is affecting your local community?"
  ) +
  coord_flip()
```



1. Lastly, try one more (different) confidence level. First, state how you expect the width of this interval to compare to previous ones you calculated. Then, calculate the bounds of the interval using the **infer** package and data from **samp** and interpret it. Finally, use the app to generate many intervals and calculate the proportion of intervals that are capture the true population proportion.

Insert your answer here In this simulation where I used a 0.99 confidence level, I expected the width of intervals to be even more narrow than with the 0.85 and 0.95 confidence levels I used earlier . The simulation

confirmed my guess.

1. Using the app, experiment with different sample sizes and comment on how the widths of intervals change as sample size changes (increases and decreases).

Insert your answer here The larger the sample size, the narrower the the widths of intervals and the smaller the sample size, the wider the widths of intervals.

sample size 15

Sample size 25

sample size 35

sample size 75

sample size 100

1. Finally, given a sample size (say, 60), how does the width of the interval change as you increase the number of bootstrap samples. **Hint:** Does changing the number of bootstrap samples affect the standard error?

Insert your answer here No, it does not.