Confidence Intervals

Zachary Palmore

2020-10-04

```
library(tidyverse)
library(DATA606)
```

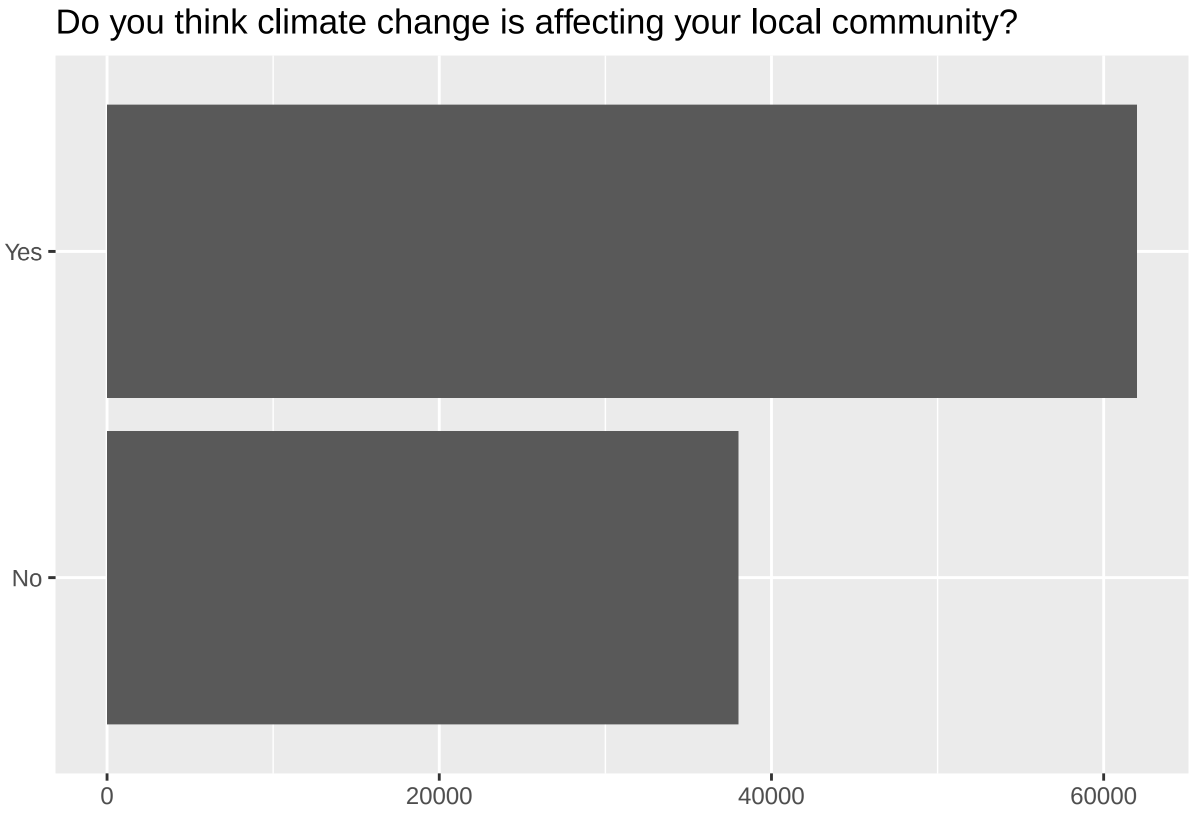
```
##
## Welcome to CUNY DATA606 Statistics and Probability for Data Analytics
## This package is designed to support this course. The text book used
## is OpenIntro Statistics, 3rd Edition. You can read this by typing
## vignette(’os3’) or visit www.OpenIntro.org.
##
## The getLabs() function will return a list of the labs available.
##
## The demo(package=’DATA606’) will list the demos that are available.
```

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

Pre-exercise

Creating the population data from lab code.

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



```
us_adults %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
```

Creating the sample from the population data.

```
n <- 60
set.seed(1050)
samp <- us_adults %>%
  sample_n(size = n)
```

Exercise 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.

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

In my sample, 55% of people think climate change affects their local community. As a proportion, that is 0.55 of the data while 0.45 or 45% responded with “No”, they did not think climate change affects their local community.

Exercise 2

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

Another student's sample proportion should be similar but not identical. This is because we are randomly sampling from the same population. The random selection creates variations in the data collected and thus the proportions calculated. However, the pool of information we are collecting the data from all have similar characteristics and will produce similar results to the population proportion as a whole.

We can create other students' examples to demonstrate.

```r
set.seed(1051)
samp1 <- us_adults %>%
  sample_n(size = n)
samp1 %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
set.seed(1052)
samp2 <- us_adults %>%
  sample_n(size = n)
samp2 %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
set.seed(1053)
samp3 <- us_adults %>%
  sample_n(size = n)
samp3 %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
set.seed(1054)
samp4 <- us_adults %>%
  sample_n(size = n)
samp4 %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
set.seed(1055)
samp5 <- us_adults %>%
  sample_n(size = n)
samp5 %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
```

Where the sample size, n, remains 60, we see little variations in the other students' proportions, but all are close in value to the true population proportions of 0.38 who think climate change does not affect their local community and 0.62 who do think climate change affects their local community.

Exercise 3

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

Example of the 95% confidence interval:

```r
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    0.417    0.667
```

The 95% confidence interval is the range of data of which you can be 95% confident contains the true value of interest. The true value is that of the population as a whole.

Exercise 4

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?

Reusing code from above:

```r
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    0.433    0.667
```

Given that the true population proportion is 0.62 for those adults that think climate change affects their local community, yes, this confidence interval captures it. The confidence interval created has a lower bound of 0.43, and an upper bound of 0.67 which includes 0.62.

Exercise 5

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?

If we ran the sample and confidence interval again on another student's sample (for example samp1) we should see some changes.

```r
samp1 %>%
  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    0.483    0.733
```

We notice the confidence interval did change. This is because the data collected in the sample was also changed during the random sampling process for samp1. Though we could repeat this and find the confidence interval of many other students' samples we would find that about 95% of those students' intervals should include the true population mean. For example, if we had 25 students with samples, around 24 of their confidence intervals might include the true population proportion.

Exercise 6

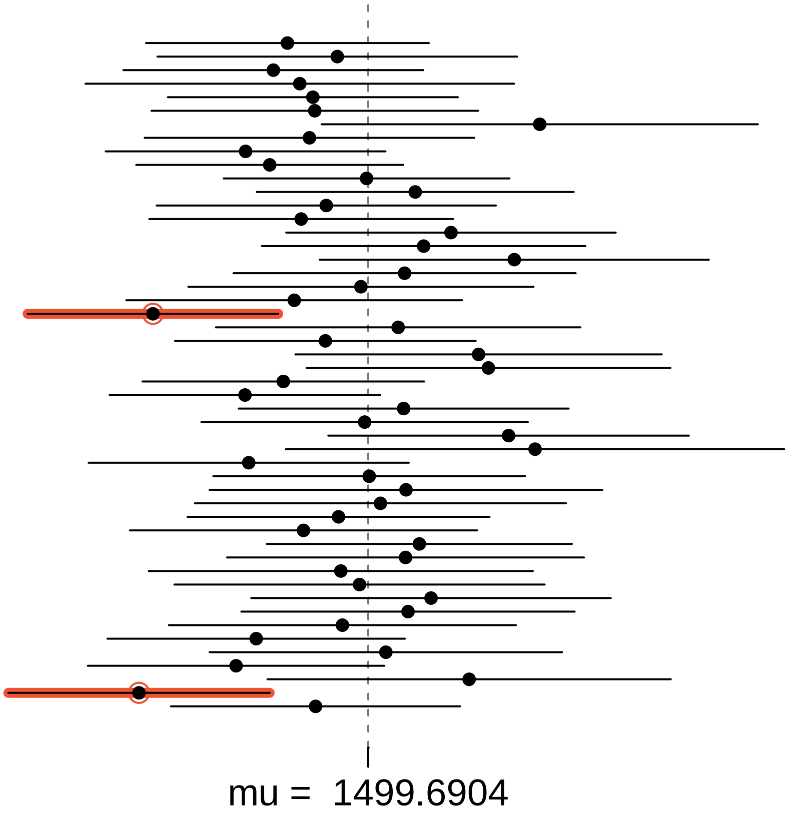
Given a sample size of 60, 1000 bootstrap samples for each interval, and 50 confidence intervals constructed (the default values for the above app), what proportion of your confidence intervals include the true population proportion? Is this proportion exactly equal to the confidence level? If not, explain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

```
# Referencing this app?
download.file("http://www.openintro.org/stat/data/ames.RData", destfile = "ames.RData")
load("ames.RData")
population <- ames$Gr.Liv.Area
set.seed(021)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
q <- 60
set.seed(10013)
for(i in 1:500){
  sampleP <- sample(population, q)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
lower_vector <- sampleP_mean - 1.96 * sampleP_sd / sqrt(q)
upper_vector <- sampleP_mean + 1.96 * sampleP_sd / sqrt(q)
```

```
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1355.575 1566.925
```

```
# Plotting as the app would generate
plot_ci(lower_vector, upper_vector, mean(population))
```



```
48/50
```

```
## [1] 0.96
```

In the first run, the sample produced 2 intervals that did not include the true population mean of 1499.69. However, the remaining 48 did produce intervals that contained the true population mean. This gives us 48 intervals that correctly captured the population mean out of 50 total intervals or a probability of 0.96 (or 96%) in the first sampling. This sampling process was run multiple times after collecting this first result and gave different results in the new samplings. Each was no more than 4% from the confidence interval specified of 95%.

Therefore we could say approximately 95% of the confidence intervals constructed include a true population confidence level. The first proportion of 96% was not exactly the same as the confidence level because there is variation in the process through random selection of variables (such as the samples selected) that is not likely to produce the same results. It is far more likely that the interval will be close to or similar to the confidence interval but not identical.

Exercise 7

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.

At a confidence level of 99% I would expect the interval to be wider than the confidence interval calculated at the 95% interval. This is due to the probability that makes the interval true when there are variations in random sampling. By setting confidence to 99% (compared to 95%), more data must be included to ensure the interval contains the true population parameter.

Here, the confidence interval is set to 0.99 or 99%. We can compare to the original 0.95 or 95% confidence interval on the same sample.

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.99)
```

To compare, the original confidence intervals were .38 and .62 at 95% confidence. Now, at 99% confidence we have 0.38 and 0.72. This makes the 99% confidence interval a wider interval than 95%.

As the authors of *OpenIntro to Statistics* stated, it is like fishing. You need a bigger net (or higher confidence interval) to be sure you are going to catch the fish you want.

Exercise 8

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.

Based on the sample and using a confidence interval of 90%;

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.90)
```

We can be 90% confident that 45% to 65% of U.S. adults think climate change is affecting their local community.

Exercise 9

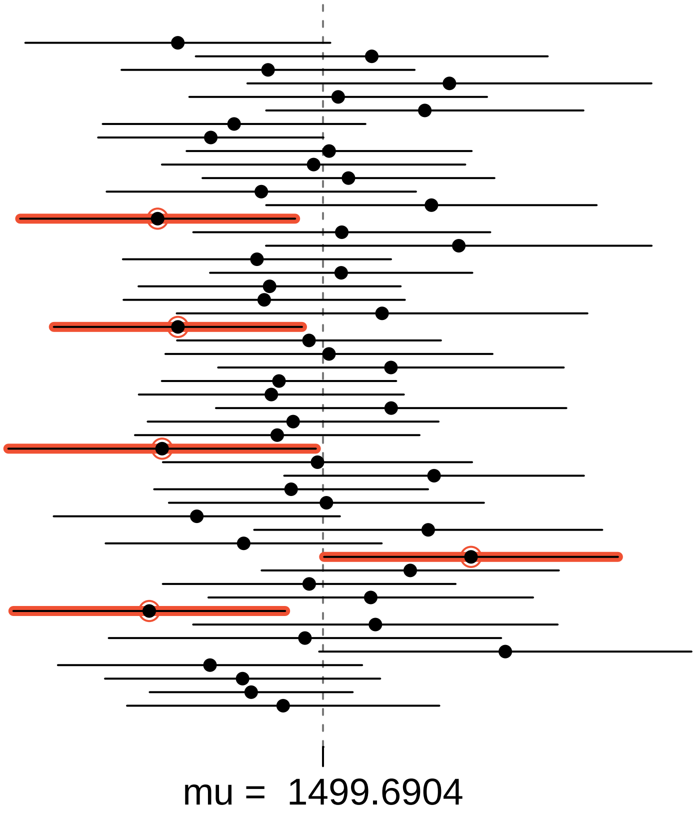
Using the app, calculate 50 confidence intervals at the confidence level you chose in the previous question, and plot all intervals on one plot, and calculate the proportion of intervals that include the true population proportion. How does this percentage compare to the confidence level selected for the intervals?

Calculating 50 confidence intervals at the confidence level of 90%, plotting all intervals on one plot and calculating the proportion of intervals that include the true population.

```
set.seed(031)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
n <- 60
set.seed(10015)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 90% CI by z-value 1.645
lower_vector <- sampleP_mean - 1.645 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 1.645 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1364.592 1579.875
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



Total number of confidence intervals that include the true population mean is 45 out of 50.

```
45/50
```

```
## [1] 0.9
```

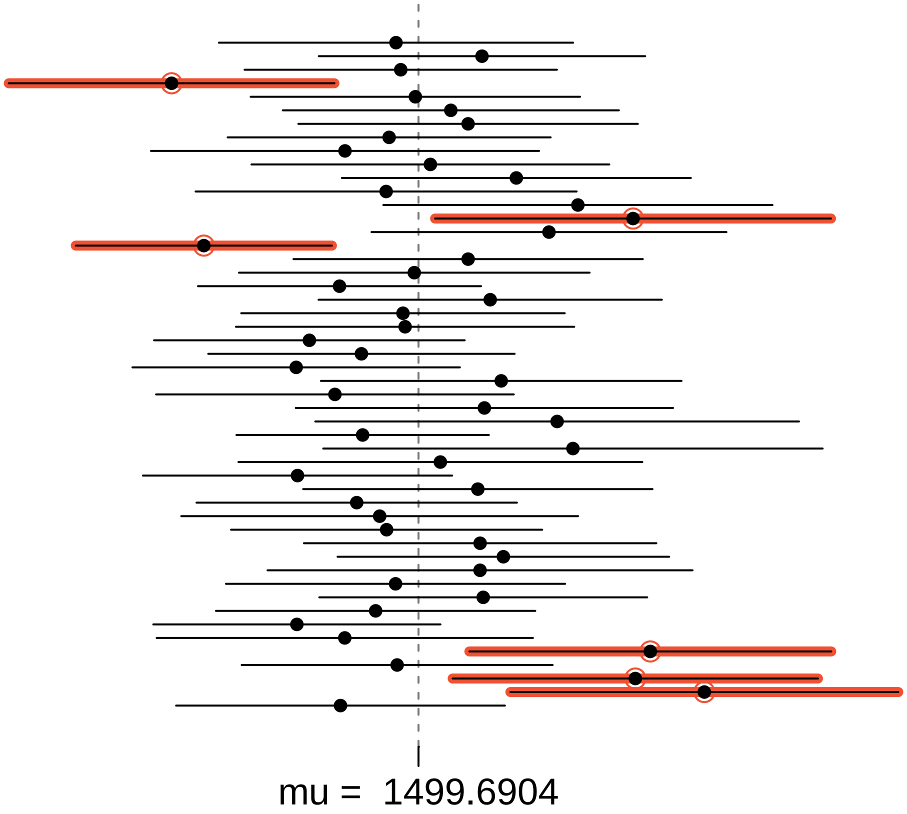
Written a proportion, that is 0.9 or 90% of the data. This percentage just happens to be exactly the percentage of the confidence interval but this may not always be the case. Let's go for one more trial.

```
set.seed(032)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
```

```
sampleP_sd <- rep(NA, 500)
n <- 60
set.seed(10016)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 90% CI by z-value 1.645
lower_vector <- sampleP_mean - 1.645 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 1.645 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1351.668 1552.466
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



This time, we ran the samples, calculations, plotted it all on one plot and found that 6 intervals did not capture the true population mean. This leaves us with a new proportion of 44 intervals that captured the true population mean out 50 total intervals.

```
44/50
```

```
## [1] 0.88
```

That can shown as 88% of the intervals plotted captured the true population value. This is not identical to the 90% confidence interval we selected but is close at only 2% away from 90%.

Exercise 10

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 capture the true population proportion.

Using the confidence interval 50% I expect the width of this interval to be smaller than all previous intervals.

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

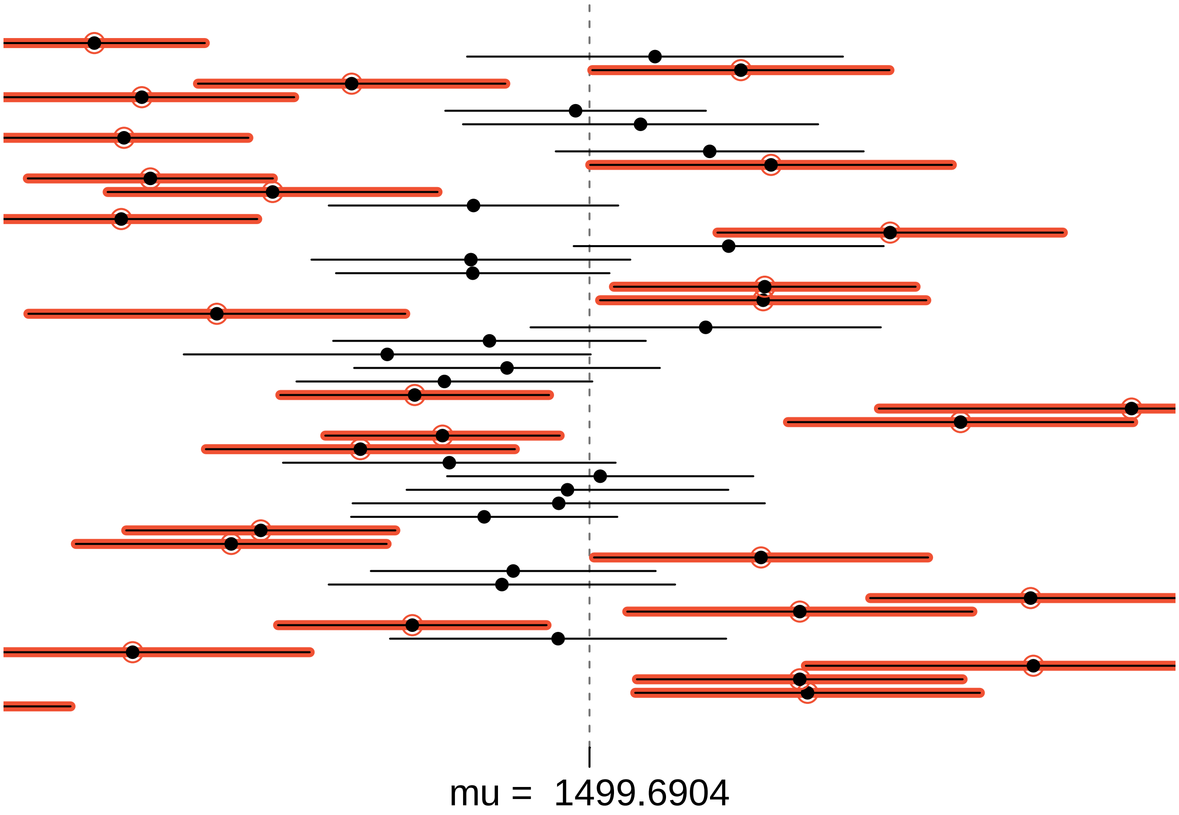
I am 50% confident that the proportion of U.S. adults that think climate change is affecting their local community is between 50% and 60%.

Using the app;

```
set.seed(040)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
n <- 60
set.seed(1023)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 50% CI by z-value 0.67449
lower_vector <- sampleP_mean - 0.67449 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 0.67449 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1289.598 1362.069
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



From this we can see that 29 intervals missed the true population mean. That leaves us with 21 intervals that did capture it. We can express this as a proportion of 0.42.

```
21/50
```

```
## [1] 0.42
```

Exercise 11

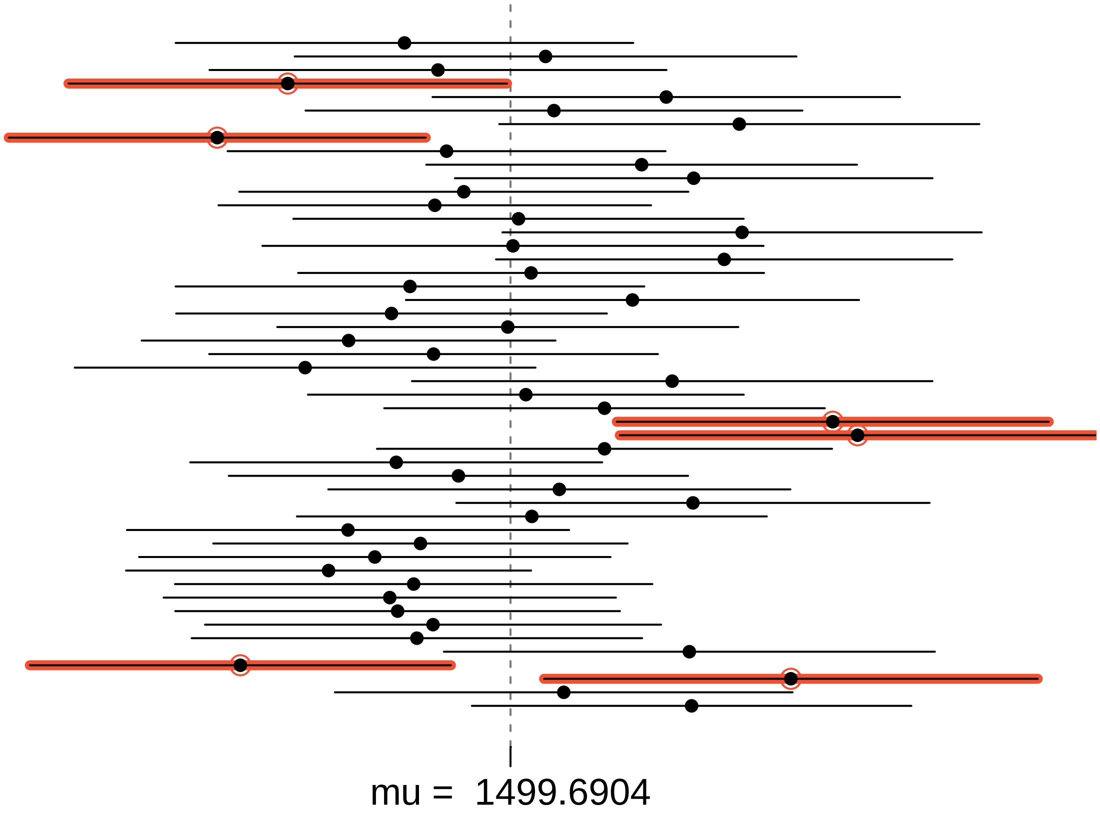
Using the app, experiment with different sample sizes and comment on how the widths of intervals change as sample size changes (increases and decreases).

As sample size increases the confidence interval narrows. As sample size decreases, the confidence interval widens. This is due to the greater level of uncertainty with smaller sample sizes. The larger the sample, the more accurate the estimate and therefore, the smaller the interval needs to be to ensure its level of confidence.

```
set.seed(054)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
n <- 400
set.seed(19016)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 90% CI by z-value 1.645
lower_vector <- sampleP_mean - 1.645 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 1.645 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1492.724 1571.931
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



With the sample size at 400 we get a proportion of 0.88, or 88%. The intervals tighten themselves compared to the first 90% CI at a sample size of 60.

```
(50-6)/50
```

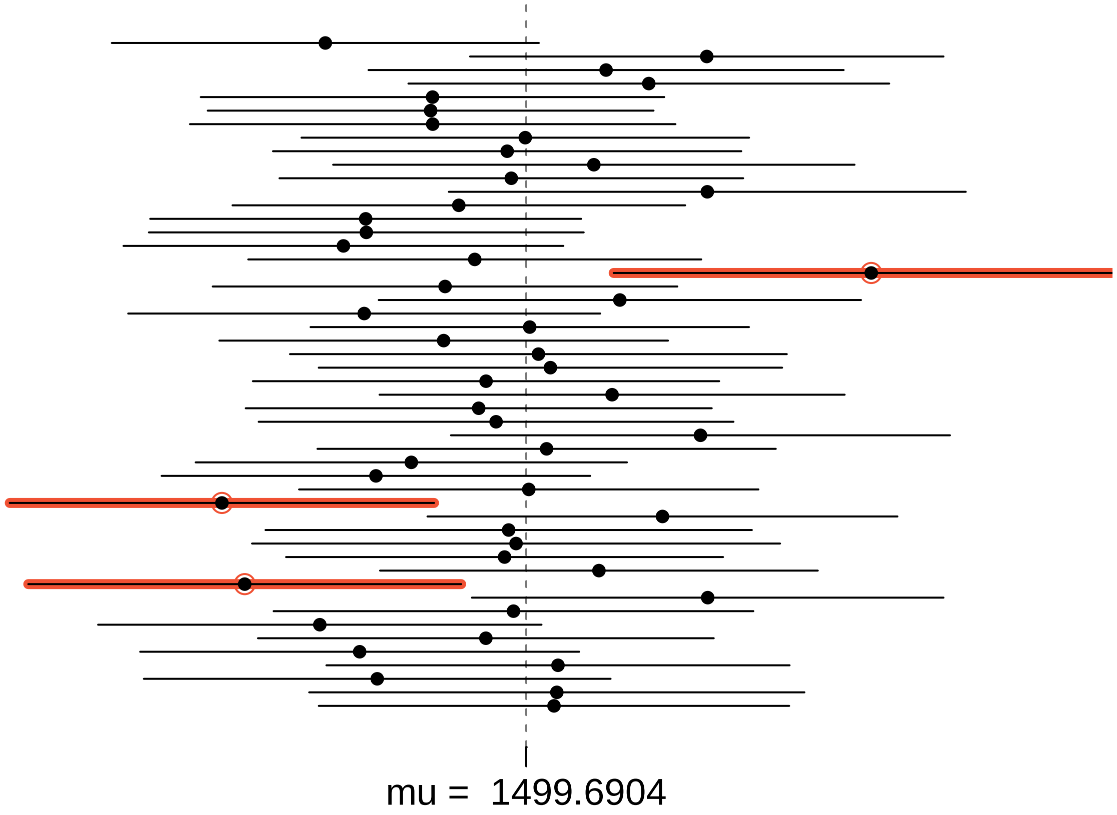
```
## [1] 0.88
```

```
set.seed(064)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
```

```
n <- 499
set.seed(190216)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 90% CI by z-value 1.645
lower_vector <- sampleP_mean - 1.645 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 1.645 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1466.942 1541.206
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



The proportion here remains close but is the best capture probability yet at 94%. The intervals seemed to close in slightly more than 400.

```
(50-3)/50
```

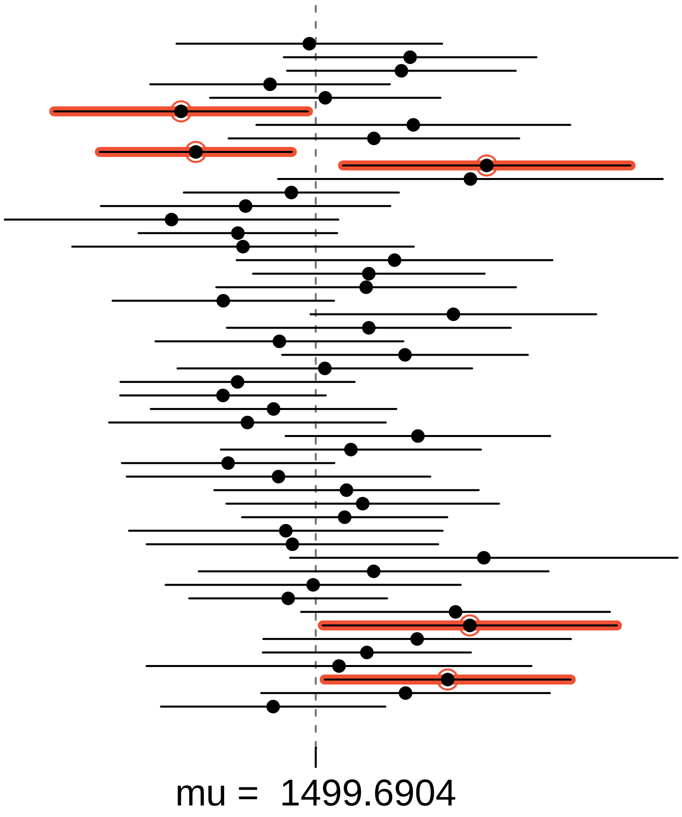
```
## [1] 0.94
```

Lastly, a samller number than the original sample size of 60 at 90% CI.

```
set.seed(074)
samppleP <- sample(population, 60)
sampleP_mean <- rep(NA, 500)
sampleP_sd <- rep(NA, 500)
n <- 30
set.seed(19836)
for(i in 1:500){
  sampleP <- sample(population, n)
  sampleP_mean[i] <- mean(sampleP)
  sampleP_sd[i] <- sd(sampleP)
  replace = TRUE
}
# Using 90% CI by z-value 1.645
lower_vector <- sampleP_mean - 1.645 * sampleP_sd / sqrt(n)
upper_vector <- sampleP_mean + 1.645 * sampleP_sd / sqrt(n)
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1328.225 1576.841
```

```
plot_ci(lower_vector, upper_vector, mean(population))
```



It gives us a proportion of 0.9 or 90%.

```
(50-5)/50
```

```
## [1] 0.9
```

Compared to the largest sample, this plot seems very zoomed out. Though, more tests are going to be run using the interval values directly without plotting. That way we can compare magnitude of the values.

What follows are a few examples of this in action. Our previously run sample size was 60 but we start with a much smaller one for comparison.

At sample size of 20 we find the confidence intervals.

```
# sample size 20
set.seed(1101)
samp_20 <- us_adults %>%
  sample_n(size = 20)
samp_20 %>%
  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      0.6     0.95
```

Its lower and upper intervals were 0.6 and 0.95 using a 95% confidence interval.

This is much larger than our original sample size. That is because in order to be confident that the true parameter exists in the data with a smaller sample size, a bigger interval is needed to capture more of the wider distribution.

The original sample size and confidence intervals.

```
# Sample size 60
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    0.417    0.683
```

Its lower and upper intervals were 0.41 and 0.68 using a 95% confidence interval.

At twice the sample size we have the sample size set at 120.

```r
# sample size 120
set.seed(1058)
samp_120 <- us_adults %>%
  sample_n(size = 120)
samp_120 %>%
  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    0.733
```

Its lower and upper intervals were 0.56 and 0.73 using a 95% confidence interval.

At quadruple the sample size we have the sample size set at 240.

```r
# sample size 240
set.seed(1059)
samp_240 <- us_adults %>%
  sample_n(size = 240)
samp_240 %>%
  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    0.521    0.646
```

Its lower and upper intervals were 0.52 and 0.65 using a 95% confidence interval.

At 10 times the sample size we have the sample size set to 600.

```r
# sample size 600
set.seed(1060)
samp_600 <- us_adults %>%
  sample_n(size = 600)
samp_600 %>%
  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    0.570    0.647
```

Its lower and upper intervals were 0.57 and 0.65 using a 95% confidence interval.

Lastly at, 100 times the sample size, we have its sample size set at 60,000.

```
# sample size 600
set.seed(1060)
samp_6000 <- us_adults %>%
  sample_n(size = 6000)
samp_6000 %>%
  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    0.613    0.639
```

Its lower and upper intervals were 0.61 and 0.64 using a 95% confidence interval.

Exercise 12

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 bootstap samples affect the standard error?

Changing the number of bootstrap samples does not affect the width of the standard error nor the confidence interval. The number of samples is not included in the standard error formula. It only requires a sample size and either sample or population proportion. The lack of change can be demonstrated through the calculation of standard error.

```
# A reminder of the population proportion
us_adults %>%
  count(climate_change_affects) %>%
  mutate(p = n /sum(n))
```

Given that we are looking for the proportion of the population that thinks climate change does have an affect on their community (in other words, “Yes” responses), the proportion is calculated as 0.62.

```
# Standard error function
st_error <- function(n, p) {
  mth <- p*(1-p)
  SE <- sqrt(mth/n)
  return(SE)
}

stats <- as.data.frame(c(20, 60, 120, 240, 600, 6000,
                         20, 60, 120, 240, 600, 6000))
stats$size <- stats$`c(20, 60, 120, 240, 600, 6000, 20, 60, 120, 240, 600, 6000)`
stats$reps <- c(1000, 1000, 1000, 1000, 1000, 1000,
                10000, 10000, 10000, 10000, 10000, 10000)
stats$p <- c(0.62)
stats$se <- st_error(stats$size, stats$p)
stats[2:5]
```

```
##    size  reps    p          se
## 1    20  1000 0.62 0.108535708
## 2    60  1000 0.62 0.062663120
## 3   120  1000 0.62 0.044309517
## 4   240  1000 0.62 0.031331560
## 5   600  1000 0.62 0.019815819
## 6  6000  1000 0.62 0.006266312
## 7    20 10000 0.62 0.108535708
## 8    60 10000 0.62 0.062663120
## 9   120 10000 0.62 0.044309517
## 10  240 10000 0.62 0.031331560
## 11  600 10000 0.62 0.019815819
## 12 6000 10000 0.62 0.006266312
```

Using those same samples we can also demonstrate this lack of change in the confidence intervals.

Each run will use a bootstrap rep of 10,000 instead of 1,000 starting with the sample size of 20;

```
samp_20 %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1      0.6     0.95
```

We see the intervals remained the same at 0.6 and 0.95.

For sample size of 60;

```
samp %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.417    0.667
```

We again see no change in the intervals of 0.42 and 0.68.

For a sample size of 120;

```
samp_120 %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.567    0.733
```

The intervals are 0.57 and 0.73 which reflects no change.

For a sample size of 240;

```
samp_240 %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.525     0.65
```

The intervals are 0.53 and 0.65 which reflects little to no change.

For a sample size of 600;

```
samp_600 %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.570    0.647
```

The intervals are 0.57 and 0.65 which reflects no change.

For a sample size of 6000;

```
samp_6000 %>%
  specify(response = climate_change_affects, success = "Yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    0.614    0.638
```

The intervals are 0.61 and 0.64 which reflects no change.

…