Simulation-based inference - confidence intervals

Ella van Engen

March 3, 2021

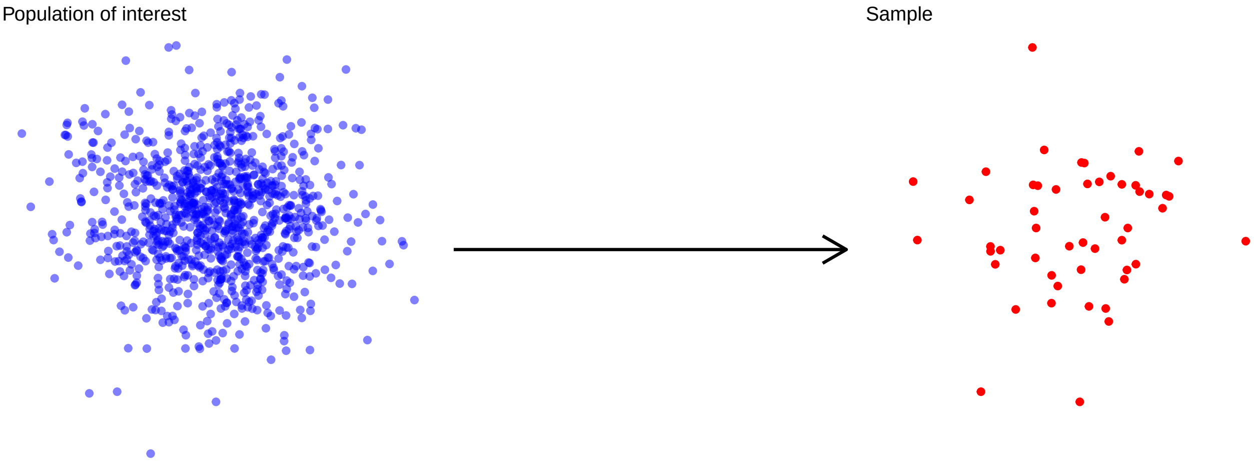
Main ideas

- Understand sample statistic variability
- Bootstrap idea and how to use it
- Generating bootstrap confidence intervals
- Correctly interpret confidence intervals

Packages

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

Big picture



Terminology

Population: a group of individuals or objects we are interested in studying

Parameter: a numerical quantity derived from the population (almost always unknown)

Sample: a subset of our population of interest

Statistic: a numerical quantity derived from a sample

Common population parameters of interest and their corresponding sample statistic:

Quantity	Parameter	Statistic
Mean	μ	$\bar{x}$
Variance	σ^2	s^2
Standard deviation	σ	s
Median	M	$\tilde{x}$
Proportion	p	$\hat{p}$

Statistical inference

Statistical inference is the process of using sample data to make conclusions about the underlying population the sample came from.

- **Estimation**: estimating an unknown parameter based on values from the sample at hand
- **Testing**: evaluating whether our observed sample provides evidence for or against some claim about the population

Today we will focus on estimation.

Estimation

Point estimate

A point estimate is a single value computed from the sample data to serve as the "best guess", or estimate, for the population parameter.

Suppose we were interested in the population mean. What would be natural point estimate to use?

Quantity	Parameter	Statistic
Mean	μ	$\bar{x}$
Variance	σ^2	s^2
Standard deviation	σ	s
Median	M	$\tilde{x}$
Proportion	p	$\hat{p}$

What is the downside to using point estimates?

Confidence intervals

A plausible range of values for the population parameter is an interval estimate. One type of interval estimate is known as a **confidence interval**.

- If we report a point estimate, we probably won't hit the exact population parameter.
- If we report a range of plausible values, we have a good shot at capturing the parameter.

Variability of sample statistics

- In order to construct a confidence interval we need to quantify the variability of our sample statistic.
- For example, if we want to construct a confidence interval for a population mean, we need to come up with a plausible range of values around our observed sample mean.
- This range will depend on how precise and how accurate our sample mean is as an estimate of the population mean.

- Quantifying this requires a measurement of how much we would expect the sample mean to vary from sample to sample.

Suppose you randomly sample 50 students and 5 of them are left handed. If you were to take another random sample of 50 students, how many would you expect to be left handed? Would you be surprised if only 3 of them were left handed? Would you be surprised if 40 of them were left handed?

Quantifying the variability of a sample statistic

We can quantify the variability of sample statistics using

1. **simulation**: via bootstrapping (today);
2. **theory**: via Central Limit Theorem (later in the course).

Bootstrapping

- The term **bootstrapping** comes from the phrase "pulling oneself up by one's bootstraps", to help oneself without the aid of others.
- In this case, we are estimating a population parameter, and we'll accomplish it using data from only from the given sample.
- 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.

Bootstrapping scheme

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

Data for Examples and Practice Problems

Consider 20 1-bedroom apartments that were randomly selected on Craigslist Manhattan from apartments listed as "by owner".

```
manhattan <- read_csv("~/R/lecture-note-class-13-ellavanengen/manhattan.csv")
```

Consider 3-10 day survival of mice randomized to various neutron doses and streptomycin therapy or saline control.

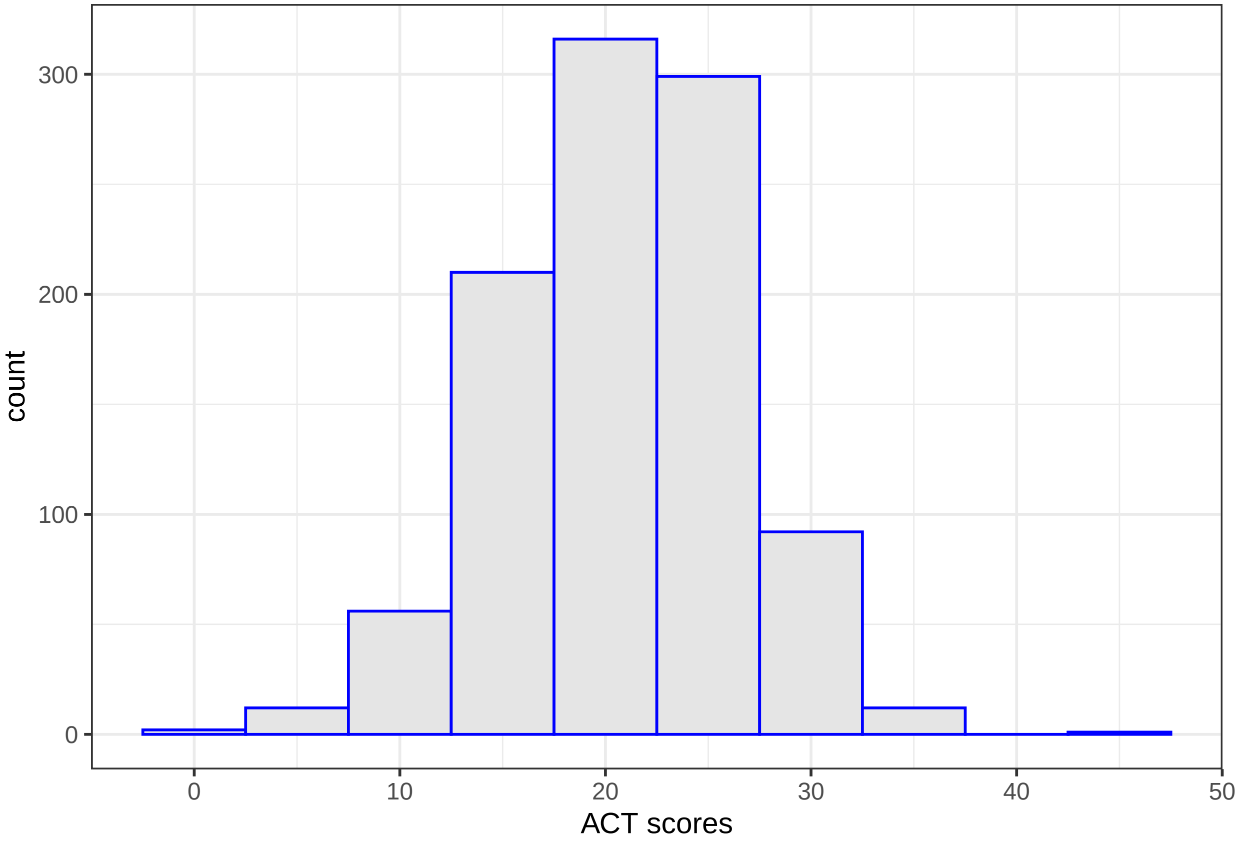
```
mice <- read_table("http://users.stat.ufl.edu/~winner/data/micerad.dat",
                   col_names = FALSE) %>%
  rename(dose = X1, treat = X2, died = X3)
```

An Example

Variability of sample statistics

Suppose the following represents the population of ACT scores. There are only 1000 individuals in our population.

```
population <- tibble(act = rnorm(n = 1000, mean = 20.8, sd = 5.8))
population %>%
  ggplot(aes(x = act)) +
  geom_histogram(binwidth = 5, fill = "grey90", color = "blue") +
  labs(x = "ACT scores") +
  theme_bw()
```



Take a sample with `slice_sample()` and compute the mean.

```
population %>%
  slice_sample(n = 30) %>%
  summarise(mean_act = mean(act))
```

```
#> # A tibble: 1 x 1
#>   mean_act
#>      <dbl>
#> 1     20.7
```

Did everyone get the same value? If fact, each time you sample (run the above code chunk) you will most likely get a slightly different mean. (Please ignore impossible values.)

The mean is not unique to this phenomenon. There will be variability for all of the sample statistics we'll be using to produce confidence intervals.

If we want to construct a confidence interval for a population mean (or any other parameter), we need to come up with a plausible range of values around our observed sample mean (statistic). This range will depend on how precise and how accurate our sample mean (statistic) is as an estimate of the population mean (parameter). Quantifying this requires a measurement of how much we would expect the sample mean (statistic) to vary from sample to sample.

Confidence intervals via bootstrap simulation

To ensure reproducibility for your knitted document, set the random number generation seed.

```
set.seed(1902370)
```

A confidence interval for the population mean - μ

Create a 95% bootstrap confidence interval for the population *mean* price of 1-bedroom apartments in Manhattan.

First, identify the following:

- Population: 1-bed apartments in Manhattan
- Parameter of interest: Price
- Sample: 1 bedroom apartments
- Sample size: 20

To create our confidence interval, we'll follow the four-step bootstrap scheme outlined in the slides using functions from `infer`.

1. **Take a bootstrap sample** - a random sample taken with replacement from the original sample, of the same size as the original sample.

2. **Calculate the bootstrap statistic** - a statistic such as mean, median, proportion, slope, etc. computed from the bootstrap samples.

3. **Repeat steps (1) and (2) many times to create a bootstrap distribution** - i.e., a distribution of bootstrap statistics.

4. **Calculate the bounds of the XX% confidence interval** as the middle XX% of the bootstrap distribution.

`infer` First, `specify()` the variable of interest, in this case `rent`.

```
manhattan %>%
  specify(response = rent)
```

```
#> Response: rent (numeric)
#> # A tibble: 20 x 1
#>     rent
#>    <dbl>
#>  1  3850
#>  2  3800
#>  3  2350
#>  4  3200
#>  5  2150
#>  6  3267
#>  7  2495
#>  8  2349
#>  9  3950
#> 10  1795
#> 11  2145
#> 12  2300
#> 13  1775
#> 14  2000
#> 15  2175
#> 16  2350
#> 17  2550
#> 18  4195
#> 19  1470
#> 20  2350
```

Second, `generate()` a fixed number of bootstrap samples. Here we'll generate 10,000 bootstrap samples. Generally, the smaller your sample size, the more bootstrap samples you'll want to generate.

```
manhattan %>%
  specify(response = rent) %>%
  generate(reps = 10000, type = "bootstrap")
```

```
#> Response: rent (numeric)
#> # A tibble: 200,000 x 2
#> # Groups:   replicate [10,000]
#>    replicate  rent
#>        <int> <dbl>
#>  1         1  2350
#>  2         1  2150
#>  3         1  3950
#>  4         1  3850
#>  5         1  1470
#>  6         1  1795
#>  7         1  2145
#>  8         1  2150
#>  9         1  3850
#> 10         1  3800
#> # ... with 199,990 more rows
```

Finally, `calculate()` the relevant statistic for each bootstrap sample.

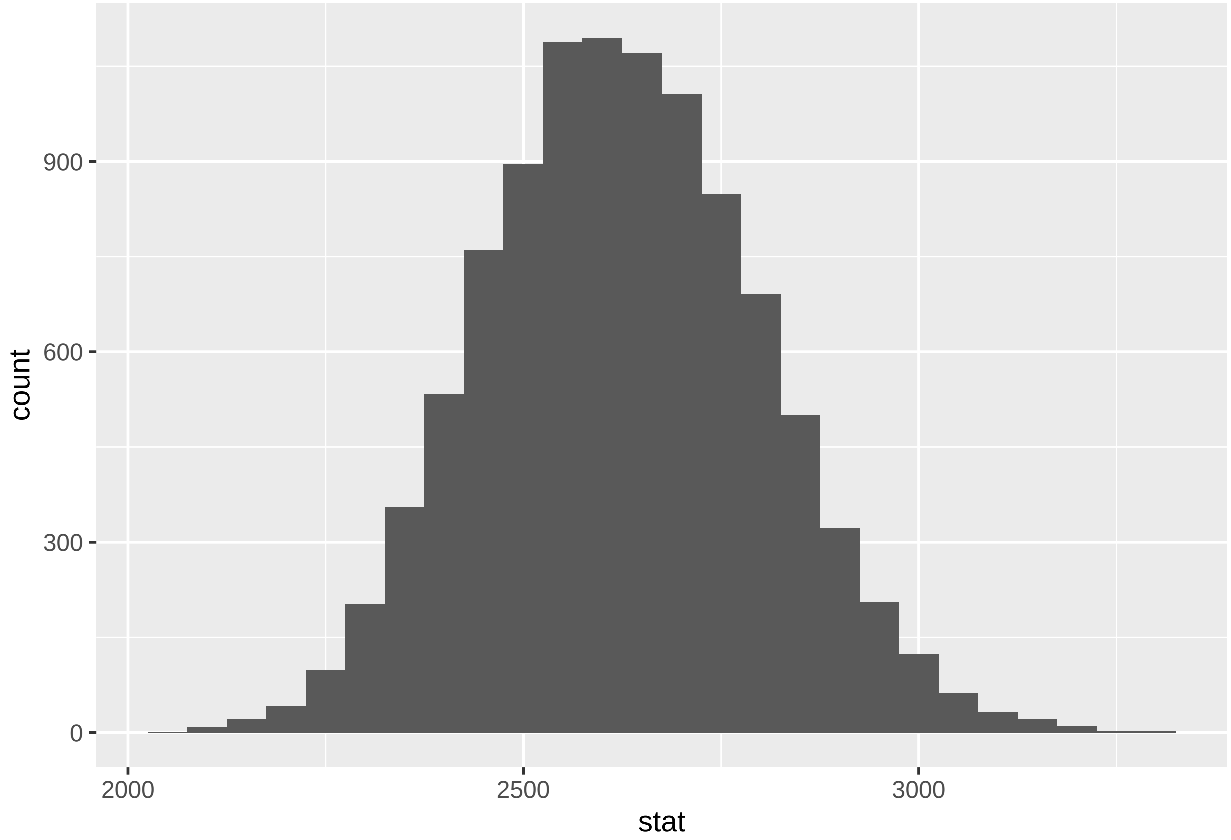
```
boot_means <- manhattan %>%
  specify(response = rent) %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "mean")
boot_means
```

```
#> # A tibble: 10,000 x 2
#>    replicate  stat
#>  *     <int> <dbl>
#>  1         1 2869.
#>  2         2 2749.
#>  3         3 2466.
#>  4         4 2738.
#>  5         5 2725.
#>  6         6 2615.
#>  7         7 2761.
#>  8         8 2788.
#>  9         9 2256.
#> 10        10 2844.
#> # ... with 9,990 more rows
```

Typically, you will do this all in one code chunk as is given in the chunk named `calculate`. It is broken up here to better understand the steps.

Visualize `boot_means` with `ggplot()`.

```
boot_means %>%
  ggplot(aes(x = stat)) +
  geom_histogram(binwidth = 50)
```



Finally, compute the lower and upper bounds to form our interval estimate. For a 95% confidence interval, these bounds occur at the 2.5% and 97.5% quantiles.

```
boot_means %>%
  summarise(
    lb = quantile(stat, 0.025),
    ub = quantile(stat, 0.975)
  )
```

```
#> # A tibble: 1 x 2
#>      lb    ub
#>   <dbl> <dbl>
#> 1 2299. 2977.
```

The package `infer` also has a convenient function to do this: `get_ci()`. Use whichever method you prefer.

```
get_ci(boot_means, level = 0.95)
```

```
#> # A tibble: 1 x 2
#>   lower_ci upper_ci
#>      <dbl>    <dbl>
#> 1    2299.    2977.
```

What does this mean?

Interpretation:

Practice Use `infer` functions to create a 95% bootstrap confidence interval for the population *median* price of 1-bedroom apartments in Manhattan. Do this in a single code chunk with a single pipeline. Provide an interpretation of your result.

```
manhattan %>%
  specify(response=rent) %>%
  generate(rep=10000, type="bootstrap") %>%
  calculate(stat="median") %>%
  get_ci(level=0.95)
```

```
#> # A tibble: 1 x 2
#>   lower_ci upper_ci
#>      <dbl>    <dbl>
#> 1    2162.     2875
```

A confidence interval for the population proportion - p

Consider the mice radiation dataset. Suppose we want to compute a 95% confidence interval for the proportion of mice that died while not on the treatment,regardless of the dose.

First, identify the following:

- Population: Mice not on treatment
- Parameter of interest: Proportion
- Sample: Mice that died while not on treatment
- Sample size: 522

```
boot_prop <- mice %>%
  filter(treat == 0) %>%
  mutate(died = if_else(died == 1, "yes", "no")) %>%
  specify(response = died, success = "yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop")
```

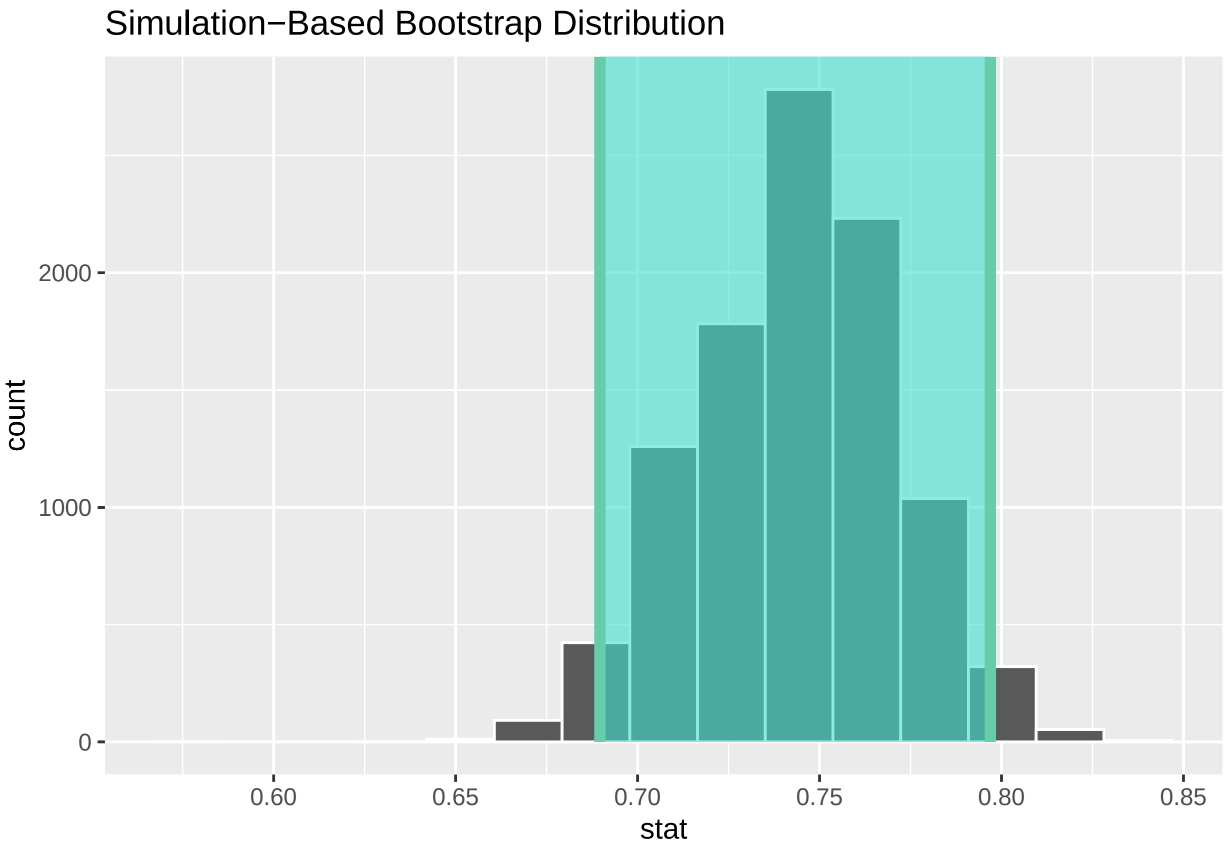
```
ci_mice_died <- get_ci(boot_prop, level = .95)
ci_mice_died
```

```
#> # A tibble: 1 x 2
#>   lower_ci upper_ci
#>      <dbl>    <dbl>
#> 1    0.690    0.797
```

What does this mean?

Interpretation:

```
visualise(boot_prop) +
  shade_ci(ci_mice_died)
```



Practice Consider the mice radiation results. Suppose we want to compute a 90% confidence interval for the proportion of mice that lived while on the treatment, regardless of the dose.

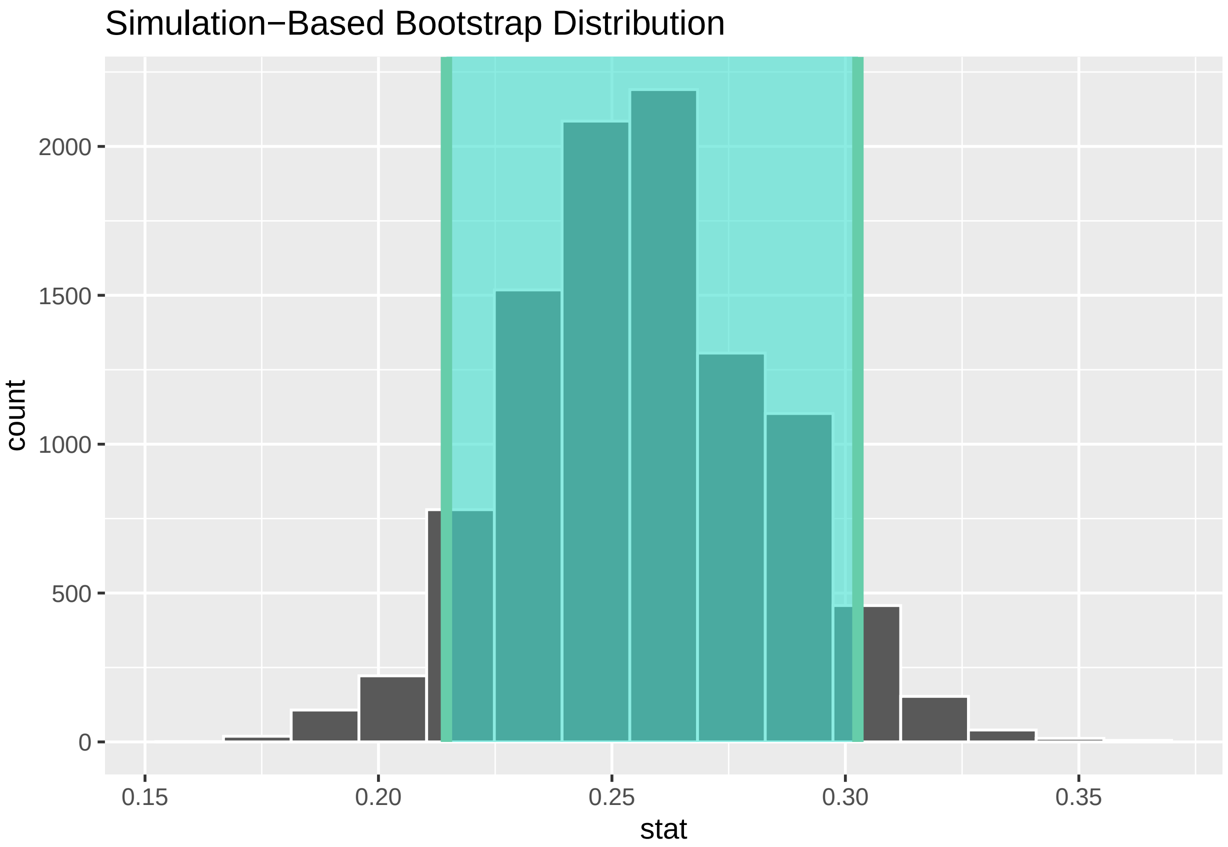
```
boot_prop <- mice %>%
  filter(treat == 0) %>%
  mutate(lived = if_else(died == 0, "yes", "no")) %>%
  specify(response = lived, success = "yes") %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "prop")
```

```
ci_mice_lived <- get_ci(boot_prop, level = .90)
ci_mice_lived
```

```
#> # A tibble: 1 x 2
#>   lower_ci upper_ci
#>      <dbl>    <dbl>
#> 1    0.215    0.303
```

Visualizing our Resuts:

```
visualise(boot_prop) +
  shade_ci(ci_mice_lived)
```



Give an interpretation of your result.

References

1. Source: C.W. Hammond, et al. (1955). "The Effect of Streptomycin Therapy on Mice Irradiated with Fast Neutrons", Radiation Research, Vol2,#4, pp.354-360