Estimation via bootstrapping

Dr. Maria Tackett

10.29.19

Click for PDF of slides

Announcements

- HW 03 **due Thursday, Oct 31 at 11:59p**

Inference

What does inference mean?

- Statistical inference is the process of using sample data to make conclusions about the underlying population the sample came from
- Types of inference: testing and estimation
- Today we discuss estimation, next time testing

Confidence intervals

Confidence intervals

A plausible range of values for the population parameter is 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

- **simulation**: via bootstrapping (today)

or

- **theory**: via Central Limit Theorem (later in the course)

Bootstrapping

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, 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.

Rent in Manhattan

How much do you think it costs to rent a typical 1 bedroom apartment in Manhattan?

Sample

On a given day, twenty 1 BR apartments were randomly selected on Craigslist Manhattan from apartments listed as "by owner".

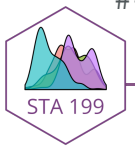
```
library(tidyverse)
manhattan <- read_csv("data/manhattan.csv")
```

```
manhattan %>% slice(1:10)
```

```
## # A tibble: 10 x 1
##     rent
##    <dbl>
##  1  3850
##  2  3800
##  3  2350
##  4  3200
##  5  2150
##  6  3267
##  7  2495
##  8  2349
##  9  3950
## 10  1795
```

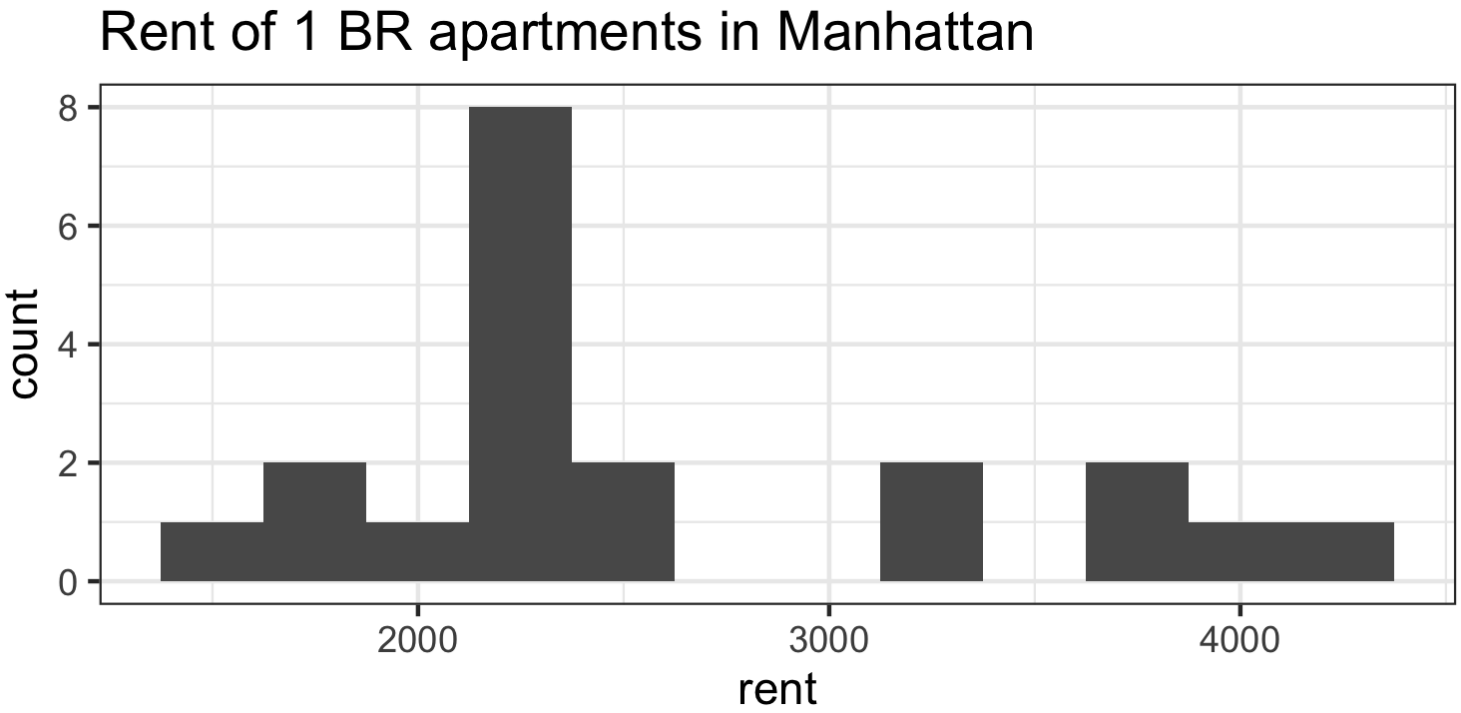
```
manhattan %>% slice(11:20)
```

```
## # A tibble: 10 x 1
##     rent
##    <dbl>
##  1  2145
##  2  2300
##  3  1775
##  4  2000
##  5  2175
##  6  2350
##  7  2550
##  8  4195
##  9  1470
## 10  2350
```



Parameter of interest

Is the mean or the median a better measure of typical rent in Manhattan?



Observed sample vs. bootstrap population

Sample median = $2350

Population median = ?

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 on 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.

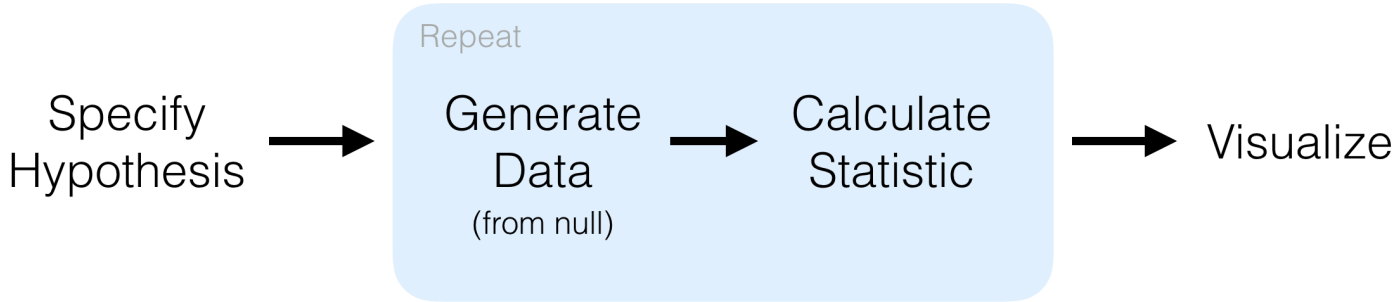
Bootstrapping in R

New package: `infer`

The objective of **`infer`** is to perform statistical inference using an expressive statistical grammar that coheres with the tidyverse design framework.

infer.netlify.com

New package: `infer`




```
library(infer)
```

Also, let's set a seed:

```
set.seed(03062019)
```

Random sampling and reproducibility

Gotta set a seed!

```
set.seed(102319)
```

- Use different seeds from each other
- Need inspiration? https://www.random.org/

Generate bootstrap medians

```
manhattan %>%
  # specify the variable of interest
  specify(response = rent)
```

Generate bootstrap medians

```
manhattan %>%
  # specify the variable of interest
  specify(response = rent)
  # generate 15000 bootstrap samples
  generate(reps = 15000, type = "bootstrap")
```

Generate bootstrap medians

```
manhattan %>%
  # specify the variable of interest
  specify(response = rent)
  # generate 15000 bootstrap samples
  generate(reps = 15000, type = "bootstrap")
  # calculate the median of each bootstrap sample
  calculate(stat = "median")
```

Generate bootstrap medians

```
# save resulting bootstrap distribution
boot_dist <- manhattan %>%
  # specify the variable of interest
  specify(response = rent) %>%
  # generate 15000 bootstrap samples
  generate(reps = 15000, type = "bootstrap") %>%
  # calculate the median of each bootstrap sample
  calculate(stat = "median")
```

The bootstrap sample

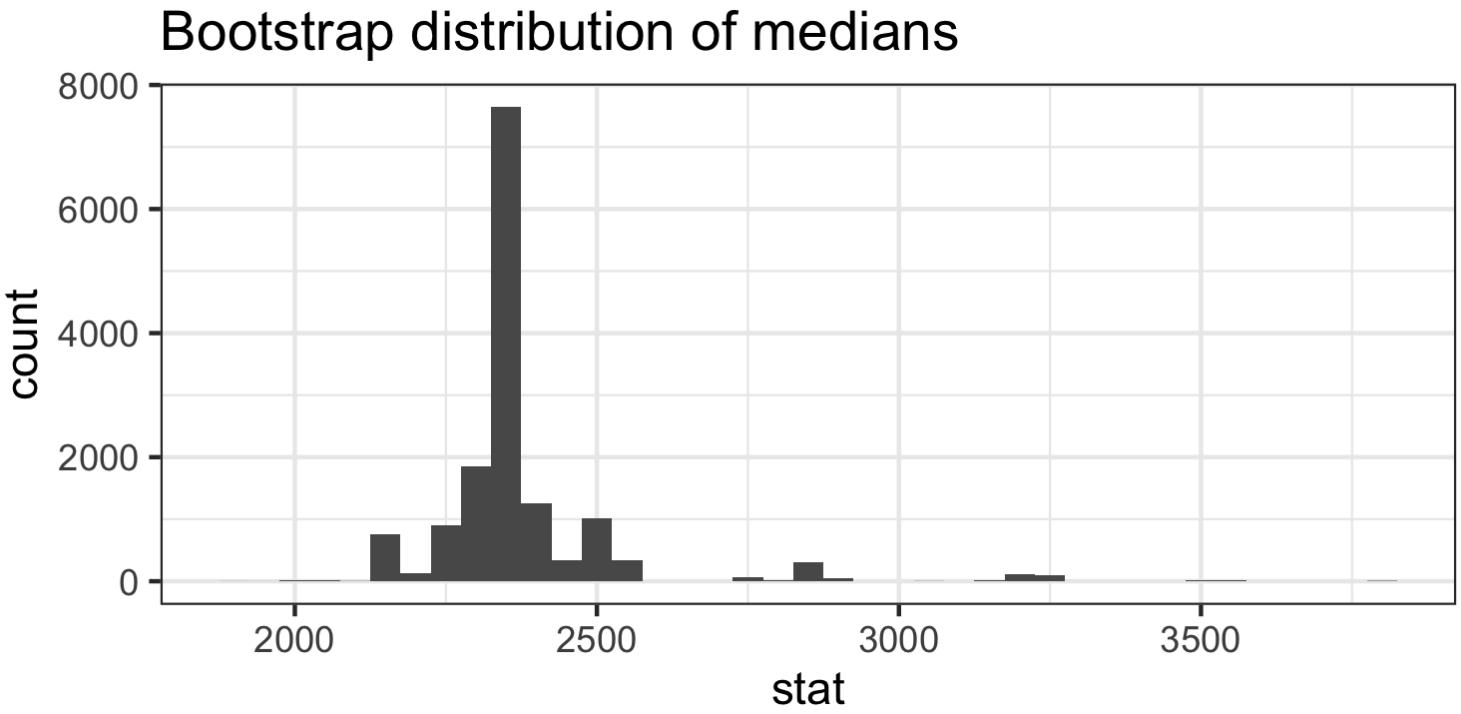
How many observations are there in **`boot_dist`**? What does each observation represent?

```
glimpse(boot_dist)
```

```
## Observations: 15,000
## Variables: 2
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,…
## $ stat      <dbl> 2350.0, 2262.0, 2350.0, 2422.5, 2262.5, 2422.5, 2237.5…
```

Visualize the bootstrap distribution

```
ggplot(data = boot_dist, mapping = aes(x = stat)) +
  geom_histogram(binwidth = 50) +
  labs(title = "Bootstrap distribution of medians")
```



Calculate the confidence interval

A 95% confidence interval is bounded by the middle 95% of the bootstrap distribution.

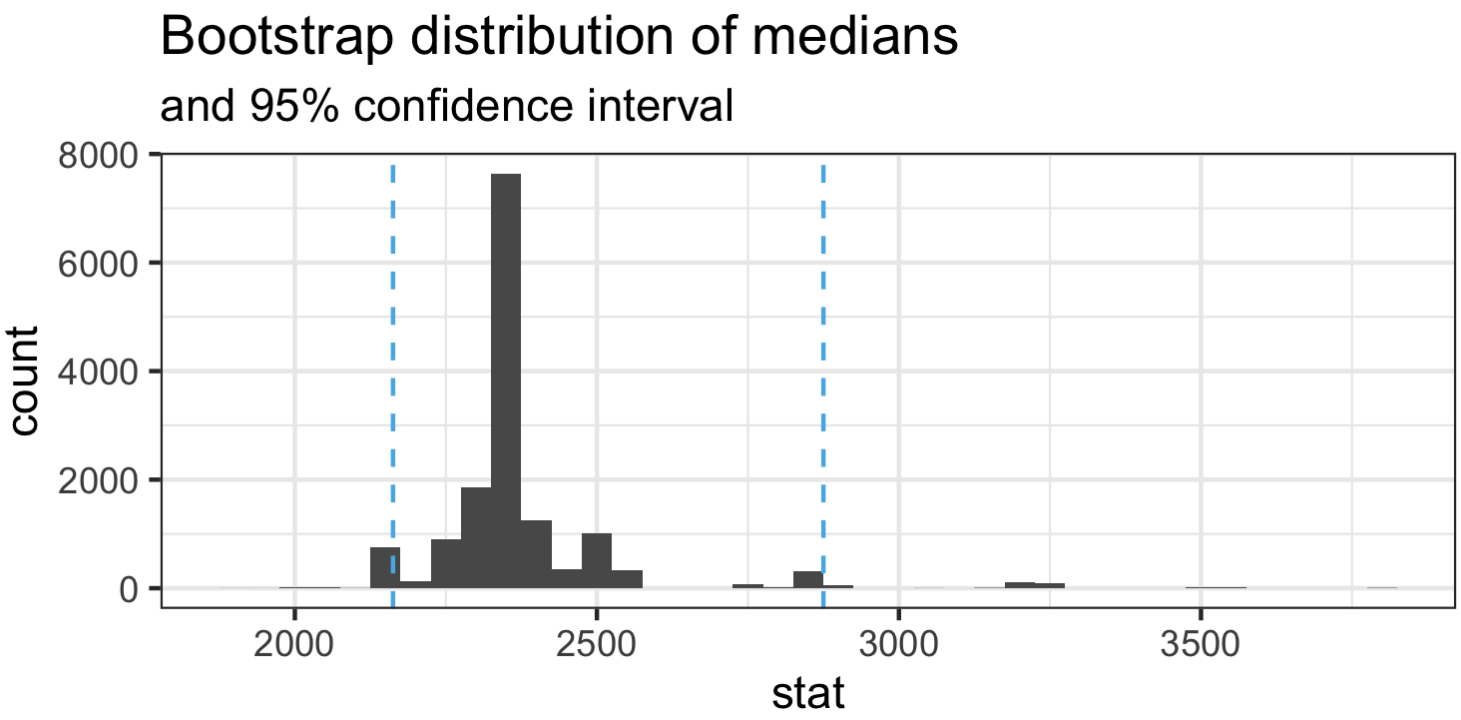
```
boot_dist %>%
  summarize(lower_bound = quantile(stat, 0.025),
            upper_bound = quantile(stat, 0.975))
```

```
## # A tibble: 1 x 2
##   lower_bound upper_bound
##         <dbl>       <dbl>
## 1       2162.        2875
```

```
(percentile_ci <- get_ci(boot_dist) )
```

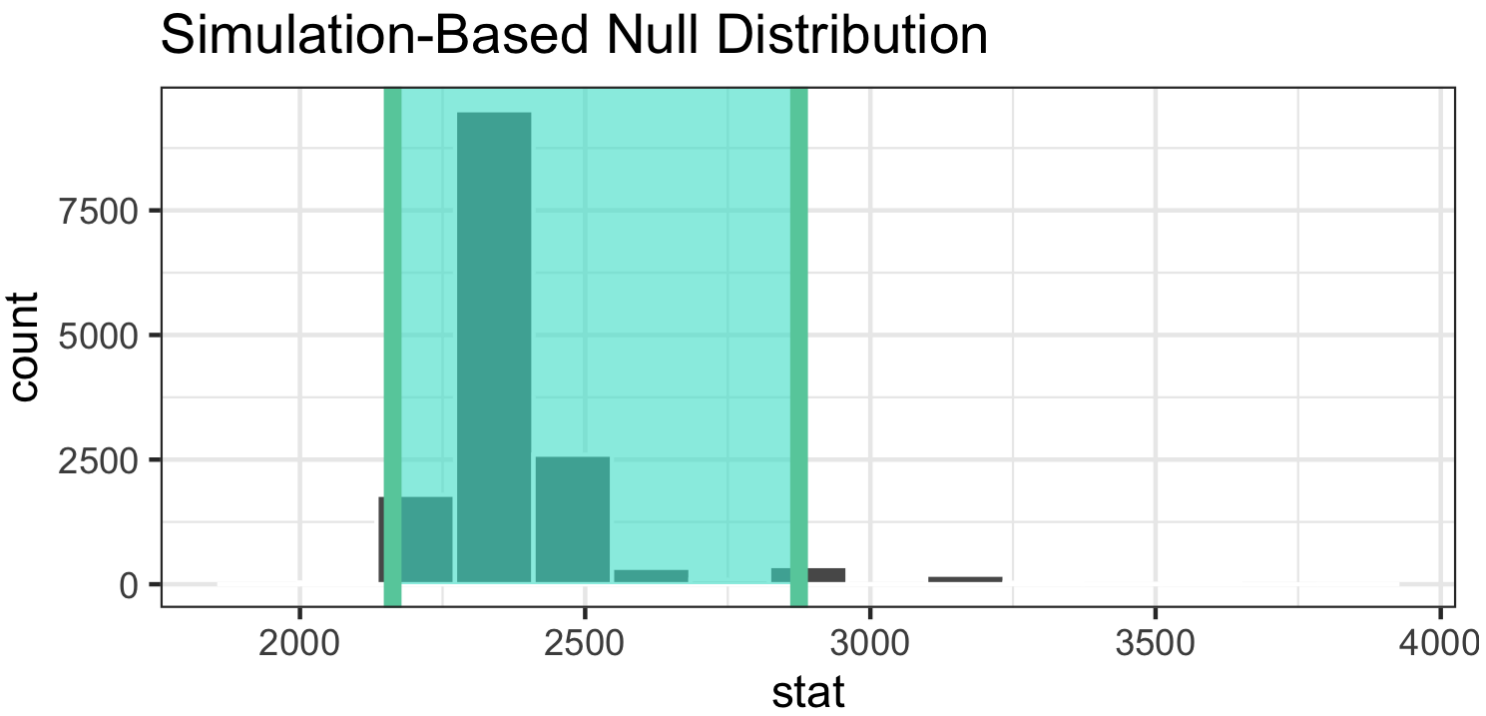
```
## # A tibble: 1 x 2
##   `2.5%` `97.5%`
##    <dbl>   <dbl>
## 1  2162.    2875
```

Visualize the confidence interval



Visualize a confidence interval

```
visualize(boot_dist) +
  shade_confidence_interval(endpoints = percentile_ci)
```



Interpret the confidence interval

The 95% confidence interval for the median rent of one bedroom apartments in Manhattan was calculated as (2162.5, 2875). Which of the following is the correct interpretation of this interval?

(a) 95% of the time the median rent one bedroom apartments in this sample is between $2162.5 and $2875.

(b) 95% of all one bedroom apartments in Manhattan have rents between $2162.5 and $2875.

(c) We are 95% confident that the median rent of all one bedroom apartments is between $2162.5 and $2875.

(d) We are 95% confident that the median rent one bedroom apartments in this sample is between $2162.5 and $2875.

Accuracy vs. precision

Confidence level

We are 95% confident that ...

- Suppose we took many samples from the original population and built a 95% confidence interval based on each sample.
- Then about 95% of those intervals would contain the true population parameter.

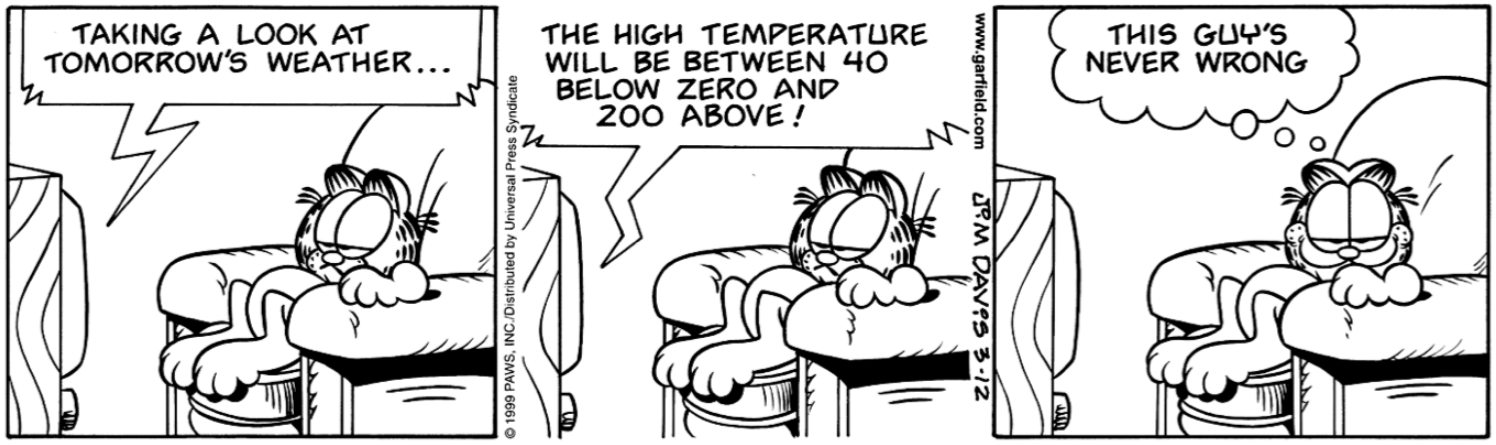
Commonly used confidence levels

Commonly used confidence levels in practice are 90%, 95%, and 99%

Which line (blue dash, green dot, orange dash/dot) represents which confidence level?

Precision vs. accuracy

If we want to be very certain that we capture the population parameter, should we use a wider interval or a narrower interval? What drawbacks are associated with using a wider interval?



How can we get best of both worlds -- high precision and high accuracy?

Calculating confidence intervals at various confidence levels

How would you modify the following code to calculate a 90% confidence interval? How would you modify it for a 99% confidence interval?

```
manhattan %>%
  specify(response = rent) %>%
  generate(reps = 15000, type = "bootstrap") %>%
  calculate(stat = "median") %>%
  summarize(lower_bound = quantile(stat, 0.025),
            upper_bound = quantile(stat, 0.975))
```

Recap

- Sample statistic ≠ population parameter, but if the sample is good, it can be a good estimate.
- We report that estimate with a confidence bound around it, and the width of this bound depends on how variable sample statistics from different samples from the population would be.
- Since we can't continue sampling from the population, we instead bootstrap from the one sample we have to estimate the sampling variability.
- We can do this for any sample statistic:
  - We did it for a median today, `calculate(stat = "median")`
  - Doing it for a mean would just take `calculate(stat = "mean")`
  - You'll learn about calculating bootstrap intervals for other statistics in lab